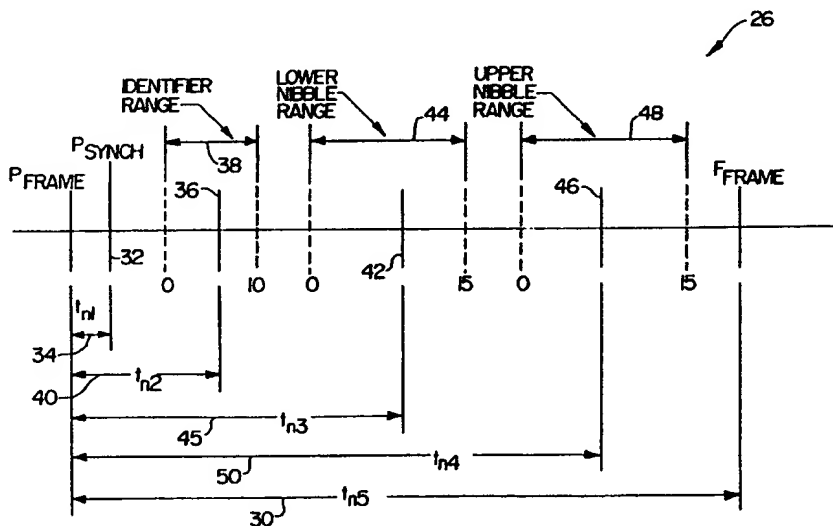




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁵ : A61N 1/08, G08C 15/06	A1	(11) International Publication Number: WO 91/10471 (43) International Publication Date: 25 July 1991 (25.07.91)
(21) International Application Number: PCT/US91/00309 (22) International Filing Date: 15 January 1991 (15.01.91) (30) Priority data: 468,407 22 January 1990 (22.01.90) US (71) Applicant: MEDTRONIC, INC. [US/US]; 7000 Central Avenue N.E., Minneapolis, MN 55432 (US). (72) Inventors: WYBORN, Paul, B. ; 1390 69th Avenue, N.E., Fridley, MN 55432 (US). ROLINE, Glenn, M. ; 4118 211th Lane, N.W., Anoka, MN 55303 (US). NICHOLS, Lucy, M. ; 6781 Marilyn Drive, Maple Grove, MN 55369 (US). THOMPSON, David, L. ; 1660 Onondaga Street, Fridley, MN 55432 (US).		(74) Agents: RISSMAN, John, A. et al.; Medtronic, Inc., 7000 Central Avenue N.E., Minneapolis, MN 55432 (US). (81) Designated States: AT (European patent), AU, BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent). Published <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: IMPROVED TELEMETRY FORMAT**(57) Abstract**

A method and apparatus are disclosed for telemetering both analog and digital data from an implantable medical device to an external receiver, such as between an implanted cardiac pacer and its external programming equipment. Analog data is first converted to digital format by an analog-to-digital converter, such that the transmission is digital data. A damped carrier at 175 kilohertz is pulse position modulated by the data. The modulation scheme defines a frame of slightly less than 2 milliseconds. The frame is divided into 64 individual time periods using a crystal clock. The data, along with synchronization and identification codes, are positioned into predefined ranges within each frame as measured by the individual time periods. The data is uniquely identified by the position of a burst of the carrier within the predefined range. This modulation scheme enables necessary data to be transmitted at sufficiently high rates with reduced power requirements thereby conserving the internal battery of the implantable device. This modulation scheme provides flexibility of use, for example, with complex medical devices where transmission of increased volumes of data is desirable, such as cardiac devices having dual-chamber or multisensor capabilities, and for controlling particular conditions, such as tachyarrhythmia.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark				

IMPROVED TELEMETRY FORMAT

BACKGROUND OF THE INVENTIONField of the Invention.

The present invention generally relates to
5 implantable medical devices, and more particularly,
pertains to telemetry schemes for percutaneously
transmitting analog and digital data from an implantable
medical device.

Description of the Prior Art.

10 The earliest implantable medical devices were
designed to operate in a single mode and with no direct
percutaneous communication. Later it became clinically
desirable to vary certain of the operating parameters and
change modes of operation. This was accomplished through
15 the use of programmers and other external devices which
transferred commands percutaneously to the implanted
medical device.

The communication between the implant and the
external world was at first primarily indirect. The
20 operation of an implantable cardiac pacer could be
observed, for example, in the electrocardiogram of the
patient. Soon it became known that data could be sent
from the implanted cardiac pacer by modulating the
stimulation pulses in some manner. This can only provide
25 a low bandpass channel, of course, without interfering
with the clinical application of the device. Change of
the pacing rate to indicate battery condition was a
commonly used application of this technique.

As implantable cardiac pacers became more complex,
30 the desirability to transfer more data at higher speeds
resulted in the percutaneous transmission of data using a
radio frequency carrier. The data to be transmitted is
of two basic types, namely, analog and digital. The
analog information can include, for example, battery
35 voltage, intracardiac electrocardiogram, sensor signals,
output amplitude, output energy, output current, and lead
impedance. The digital information can include, for

-2-

example, statistics on performance, markers, current values of programmable parameters, implant data, and patient and unit identifiers.

The earliest RF telemetry systems transmitted analog and digital information in separate formats, resulting in inefficient utilization of the available power/bandwidth. Also, these modulation schemes tended to be less than satisfactory in terms of battery consumption, and do not lend themselves to simultaneous transmission of differing data types.

Many types of RF telemetry systems are known to be used in connection with implantable medical devices, such as cardiac pacemakers. An example of a pulse interval modulation telemetry system used for transmitting analog and digital data, individually and serially, from an implanted pacemaker to a remote programmer is disclosed in U.S. Patent No. 4,556,063 issued to Thompson et al., herein incorporated by reference. An example of a modern pacemaker programmer for use with programmable cardiac pacemakers having RF telemetric capabilities is disclosed in U.S. Patent No. 4,550,370 issued to Baker, herein incorporated by reference. However, the telemetry format which is used under these systems, as well as other prior telemetry systems, have not been entirely adequate for reasons described above and a need for significant improvement has continued. As will become apparent from the following, the present invention satisfies that need.

SUMMARY OF THE INVENTION

The present invention percutaneously transmits all data from the implantable medical device in a digital format. It is pulse position modulated on an RF carrier. To accomplish this, the analog quantities must be converted to digital values either at the time of transmission, such as for real-time intracardiac electrocardiograms, or before storage in the memory of

the device, as in the case of historical values of pacing rate for subsequent transmission for trend analysis.

Whether the data to be sent is initially analog or digital, it is transmitted in the same format, i.e., as digital information. The RF carrier is pulse position modulated to conserve battery energy. In this manner, only a short burst of the carrier, e.g., one cycle, is actually needed to transmit a given unit of data. The time position of that burst relative to a synchronizing standard determines the value of the data unit transmitted.

To accomplish this pulse position modulation, a frame of about 2 milliseconds is defined. Within this frame are positioned a synchronizing burst, a frame identifier burst, and one or more data bursts. The synchronizing burst is positioned at a fixed position in the frame. The frame identifier and data are variables, such that the corresponding bursts occur within a range of time within the frame. The range in which a burst is found defines the nature or type of the variable. The position in the range defines the value of the variable.

Because all data transmission is in a digital format, great flexibility is achieved with regard to additional units of data for future applications. The use of the standardized format and capability of encoding more data into a single pulse also decreases the overall battery current requirements and serves to level the energy demand over time. Transmitting the analog data in digital form provides enhanced noise immunity and accuracy.

The transmission protocol provides data rates which are sufficient to transfer clinically useful EGM information in real time. Because each frame is independent, data quantities of varying precision can be transmitted using the same protocol. This modulation scheme provides flexibility of use, for example, with

-4-

complex medical devices where transmission of increased volumes of data is desirable in real time, such as cardiac devices having dual-chamber or multisensor capabilities, and for controlling particular conditions
5 such as tachyarrhythmia.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be better understood, and its attendant advantages will be readily appreciated, by reference to the accompanying drawings when taken in
10 consideration with the following detailed description, wherein:

FIG. 1 is a simplified schematic view of an implantable medical device employing the improved telemetry format of the present invention;

15 FIG. 2 is a conceptual view of one frame of the improved telemetry format of the present invention;

FIG. 3 is a view of the actual transmission pattern of two frames of the improved telemetry format;

FIG. 4 is a block diagram of a portion of an
20 implantable medical device for implementation of the improved telemetry format;

FIG. 5 is a simplified flowchart showing the basic function of software to perform the telemetry uplink operation of the improved telemetry format;

25 FIG. 6 is a block diagram of the circuitry of the telemetry uplink hardware for implementing the improved telemetry format;

FIG. 7 is a block diagram of the circuitry of the telemetry timing for implementing the improved telemetry
30 format; and

FIG. 8 is a schematic diagram of the driver circuitry for implementing the improved telemetry format.

-5-

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A preferred embodiment of the present invention is disclosed relating to use of the improved telemetry format with an implantable cardiac pacer, which may be
5 programmable. However, those of skill in the art will be readily able to adapt the teachings found herein to other implantable medical devices. It will also be understood by those of skill in the art that the telemetry format taught herein can be used for bi-directional
10 communications between an implanted medical device and an external device.

FIG. 1 is a simplified schematic diagram of the present invention as employed in a cardiac pacing system. An implantable pulse generator 10 is implanted in the
15 patient under the outer skin barrier 28. Implantable pulse generator 10 is electrically coupled to the heart of the patient using at least one cardiac pacing lead 12 in a manner known in the art. Percutaneous telemetry data is transmitted from implantable pulse generator 10
20 by an RF uplink 26 utilizing the improved telemetry format to a receiving antenna 22, which is coupled to a programmer 20 via a cable 24. Receiving antenna 22 also contains a magnet which activates a reed switch in implantable pulse generator 10 as a safety feature, as
25 taught in U.S. Patent No. 4,006,086 issued to Alferness et al., herein incorporated by reference. The telemetry data is demodulated and presented to the attending medical personnel by programmer 20.

FIG. 2 is a schematic diagram of the protocol of RF
30 uplink 26 using the improved telemetry format. The uplink uses a damped 175 kilohertz RF carrier which is pulse position modulated, as described in detail below. Shown at 30, the basic timing unit of the format is a frame, having a duration of t_{fs} . It will be understood by
35 those skilled in the art, however, that the present invention can be practiced using fixed-length frames

having periods of shorter or longer duration. In the preferred embodiment, the main timing source of implantable pulse generator 10 comprises a standard 32.768 kilohertz crystal clock which provides a basic 5 clock cycle of 30.52 microseconds. Thus, a frame comprised of 64 clock cycles and extending over a fixed time interval of 1.953125 milliseconds is a convenient frame period, since such frame period is a binary multiple of the basic clock cycle.

10 A unique synchronizing signal is positioned within a first fixed range of each frame 30. This signal comprises a synchronizing RF pulse 32 which is located at a time t_{n1} within frame 30. To properly function as a synchronizing pulse, it must be located at a fixed point 15 within the first fixed range of frame 30, as shown at 34.

A four-bit frame identifier code is positioned within a second fixed range of each frame 30, such second fixed range comprising an identifier range 38. Identifier range 38 uses a total of eleven basic clock 20 cycles as shown. This identifier code comprises an identifier RF pulse 36 which is pulse position modulated within the identifier range 38. The position of identifier pulse 36 within identifier range 38 identifies the nature or type of data found within each frame 30 25 which is being transmitted, such as peak sense, peak pressure, sense threshold and others, as described in further detail below. Shown at 40, time interval t_{n2} thus uniquely represents the value of identifier pulse 36, which value in turn identifies the data type being 30 transmitted within frame 30.

Each frame 30 transfers one eight-bit byte of data along with the identifier code. This data is divided into two portions comprised of four bits of data each. A first portion of this data, namely the four least 35 significant bits of the data byte, is positioned within a third fixed range of frame 30, such third fixed range

-7-

comprising a lower nibble range 44. A second portion of this data, namely the four most significant bits of the data byte, is positioned within a fourth fixed range of frame 30, such fourth fixed range comprising an upper 5 nibble range 48.

A lower nibble pulse 42 is pulse position modulated within lower nibble range 44, such that its value is uniquely identified by its location, such as at a time t_{n3} shown at 45. An upper nibble pulse 46 is also pulse 10 position modulated within upper nibble range 48, such that its value is uniquely identified by its location, such as at a time t_{n4} shown at 50. Lower nibble range 44 and upper nibble range 48 each comprise sixteen basic clock cycles, permitting each of the sixteen unique 15 values of the four-bit nibble to be specified. To prevent data overlap, suitable guardbands are positioned between each of the ranges within the frame to uniquely identify the synchronizing pulses, thereby avoiding undefined and erroneous data transmission.

20 FIG. 3 is a diagram of two frames of RF uplink 26, wherein a first frame corresponds to Word 1 shown at 70, and a second frame corresponds to Word 2 shown at 72. A count of clock cycles is indicated along an upper horizontal axis of this diagram for each frame. Each 25 basic clock cycle has a duration of 30.52 microseconds. The first frame at 70 is initiated by an RF pulse 52. A synchronizing RF pulse 54 is shown uniquely identified as precisely four clock cycles later. Because the guardbands are all greater than four clock cycles, no 30 combination of a frame identifier and data can appear as a synchronizing pulse. Synchronizing pulse 54 is used to provide frame synchronization between the transmitter (i.e., implantable pulse generator 10) and the receiver (i.e., programmer 20).

35 An identifier RF pulse 56 is located within identifier range 38, which range is defined as nine to

nineteen basic clock cycles from the beginning of frame 70. In Word 1, for example, identifier pulse 56 is located at clock cycle nineteen. This identifies the frame as a particular type of data transfer, namely,
 5 "Sense Threshold" as indicated in Table 1 below.

TABLE 1

<u>Position</u>		<u>Identification</u>
10	9	Memory
	10	Idle
	11	EGM-1
	12	Markers
	13	Peak Sense
15	14	Pressure Waveform
	15	Peak dp/dt
	16	Peak Pressure
	17	Delta Capacitor Voltage
	18	Activity Counts
	19	Sense Threshold
20		

A lower nibble RF pulse 58 is located within lower nibble range 44, which range is defined as twenty-four to thirty-nine basic clock cycles from the beginning of frame 70. In Word 1, for example, lower nibble pulse 58
 25 is located at clock cycle thirty-one, specifying a binary value of seven on a scale of zero to fifteen. An upper nibble RF pulse 60 is located at clock cycle fifty-eight within upper nibble range 48, which range is defined as forty-four to fifty-nine basic clock cycles from the
 30 beginning of frame 70, and is demodulated in similar fashion.

FIG. 4 is a block diagram of that portion of implantable pulse generator 10 which is associated with formatting and transmission of RF uplink 26. Most of the

-9-

unique hardware which embodies the present invention is located on a single substrate, being a custom chip device indicated generally by arrow 105. The remainder is microprocessor-based logic indicated generally by arrow 5 100, comprising microprocessor 102, random access memory (RAM) 104, and parallel bus 106. The function of microprocessor-based logic 100 is described in further detail below.

Chip 105 has an analog-to-digital (A/D) converter 10 108 which receives a number of analog inputs 110 from a multiplexer (not shown). A/D converter 108 permits data to be transferred via RF uplink 26 to be digitized as necessary, so that all data is transmitted in a standardized digital form.

15 Circuitry (CRC) for generating and analyzing the cyclic redundancy code used to forward error detect telemetry data transmitted over RF uplink 26 is indicated at 112. In the preferred embodiment, it is also used for data received by implantable pulse generator 10 via a 20 downlink (not shown). Circuitry (DMA) for providing direct memory access to RAM 104 is indicated at 114, thus permitting multiple byte transfers without constant management by microprocessor 102.

Key hardware used to implement RF uplink 26 25 comprises telemetry control and data buffer circuitry indicated generally within dashed lines at 121, which circuitry includes data buffer 116 and telemetry control 120, and up-link timing circuitry 118. Data buffer 116 includes storage for twelve bits of data. This storage 30 is partitioned into a four-bit section 119 for storage of the frame identifier code, and an eight-bit section 117 for storage of the lower nibble and upper nibble of a frame. Data buffer 116 thus stores all of the variables for one complete frame. Data buffer 116 is used to stage 35 the variables for the frame which may be received from

-10-

RAM 104, A/D converter 108, CRC 112, or elsewhere along parallel bus 106.

Telemetry control 120 consists primarily of a telemetry status register. This register stores the
5 telemetry commands and status as loaded by microprocessor 102. The contents of the register are thus used to gate the data at the proper time of the defined protocol.

Up-link timing 118 decodes the twelve bits of data stored in data buffer 116 to produce a set of timing
10 signals which key bursts of RF energy at the appropriate times to pulse position modulate the 175 kilohertz carrier. Up-link timing 118 also keys bursts of RF energy at the fixed positions within the frame corresponding to the frame-initiating pulse and the
15 synchronizing pulse.

FIG. 5 is a basic flowchart showing the overall function of the microprocessor-based logic 100. The role is essentially one of initiation of the transfer, rather than management of each detail of the transmission.
20 Software associated with RF uplink 26 is started at element 130, usually by a down-linked command to transfer data.

Element 132 schedules the requested transmission via the up-link facilities. This scheduling prioritizes
25 uplink transmission requests. Lower priority is given to continuous real time transfers, such as EGM and battery voltage, whereas higher priority is given to single occurrence transmissions of status information.

After scheduling, element 134 determines whether an
30 uplink transmission is currently in progress. If an uplink transmission is in progress, element 132 re-schedules the request.

If an uplink transmission is not in progress after scheduling, element 136 initiates the uplink transmission
35 by activating telemetry control 120. Exit is via element 138. While some additional management of the process is

-11-

required during the transmission, a description of such further details has been omitted, since it is not believed necessary to one skilled in the art to fully understand the present invention. As to the software 5 associated with the uplink transmission, however, a source code listing of the pertinent sections of such software has been attached hereto as Appendix A, and is incorporated by reference herein.

FIG. 6 is a block diagram showing the major data and 10 control signals of telemetry control and data buffer 121 (which includes data buffer 116 and telemetry control 120 shown in FIG. 4), and also of up-link timing 118. A primary function of data buffer 116, as indicated above, is the staging of the twelve variable bits of a given 15 frame which correspond to a four-bit frame identifier ID, and dual-nibble data comprising a four-bit lower nibble LN and a four-bit upper nibble UN. The data is received over an eight-bit, parallel bus 159 and can be from any one of several sources. Control lines EGMDATA at 150, 20 PRSDATA at 151, DLDMA at 153, DMADS at 155, LDANDAT at 156, ENCRIC at 161 and LDCRC at 171 specify the source. The output of A/D converter 108 of FIG. 4 is presented separately to data buffer 116 as an eight-bit parallel transfer to ADC(0-7) at 154 (see FIG. 6). The output of 25 CRC 112 is presented separately to data buffer 116 as an eight-bit parallel transfer to CRC(0-7) at 160, since those devices are located on the same substrate.

Telemetry control 120 outputs a number of control signals, including EGMGAIN at 162, RVPGAIN at 163, 30 EGMTELEN at 164, ANULON at 165, RAMULON at 166, MEMEN at 167, PRSTELN at 168, HDRCRCEN at 169 and EGMNPRS at 170. These control outputs are used to enable and control inputs to data buffer 116. The key outputs of telemetry control and data buffer 121 are TELRST at 182, which 35 resets up-link timing 118 and initiates the beginning of a frame, and a parallel data transfer at 184, which

-12-

transfers the frame identifier ID, lower nibble LN and upper nibble UN to up-link timing 118.

Up-link timing 118 receives the frame-initiating control signal TELRST at 182 and the parallel data transfer (ID, LN and UN) at 184. A primary function of up-link timing 118 is to key the transmission of 175 kilohertz RF energy at the proper times to indicate start of frame, frame synchronization, frame identifier, lower nibble and upper nibble. Timing for this function is provided by the 32.768 kilohertz crystal clock to up-link timing 118 with clock signal XTAL at 186. An output TELCLK is provided at 188 which keys the actual burst of RF carrier at the proper times.

FIG. 7 is a block diagram of up-link timing 118. A frame timing generator 202 provides the desired timing for a frame according to clock input XTAL at 186, in a manner hereinabove explained. Thus, each frame is comprised of sixty-four basic clock cycles. The process is initiated by receipt of the frame-initiating control signal TELRST at 182, which enables uplink when in a low state and disables uplink when in a high state. The initial clock cycle of a frame contains a burst of RF energy which is keyed by control signal TELCLK at 188, which is also used to trigger the start of the data decoding by an uplink word multiplexer 200.

After the proper four-bit quantity is selected (i.e., frame identifier ID first, lower nibble LN next, and upper nibble UN last), a telemetry pulse timer 204 determines the appropriate timing for a burst to be provided to frame timing generator 202, and a corresponding burst of RF energy is keyed. Each of the four-bit quantities thus results in the keying of a burst of RF energy at the appropriate time within each frame.

FIG. 8 is a circuit diagram for the drive circuit for generating the RF carrier. A control signal TELCLK at 188 provides the timing information for keying the

-13-

carrier. A non-overlap generator 220 functions as a delay device to save current by preventing output transistors 230 and 232 from conducting simultaneously. Every transition of control signal TELCLK at 188 causes one transition by non-overlap generator 220. Inverters 222, 224, 226 and 228 are scaled to provide efficient switching with sufficient drive to the gates of transistors 230 and 232. Transistors 230 and 232 drive the signal off of chip 105 to ANTDR at 234 to an antenna circuit. A tuned circuit of discrete components, capacitor 236 and coil 238, are located external to chip 105. Each transition thus causes this tuned circuit to resonate at 175 kilohertz, thereby generating one uplink burst.

While the invention has been described above in connection with the particular embodiments and examples, one skilled in the art will appreciate that the invention is not necessarily so limited. It will thus be understood that numerous other embodiments, examples, uses and modifications of and departures from the teaching disclosed may be made as to various other systems for telemetering data to and from an implantable medical device, without departing from the scope of the present invention as claimed herein.

APPENDIX A

-15-

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ***** R2 SYSTEM DATA AREA ***** File: DATA.ASH
 ***** \$Revision: 3.0 \$ *****

10/12/89 08:11:23
 Page 9

```

=0005      400  ext_tlm_active EQU 5      ;Extended telemetry is active
=0006      401  mag_state     EQU 6      ;Magnet state, mode and rate are
      402      ;set to VOO_MODE and mag_rate following
      403      ;permanent programming.
=0007      404  rr_trans      EQU 7      ;Rate response transition
=0080      405
      406  TLM_NOWMAG_MSK EQU 10110000B ;Mask to clear all telemetry
      407      ;flags except those associated
      408      ;with extended telemetry.
      409
      410  ;*****
      411  ;*          tlm2_flags
      412  ;*****
=0000      413  perm_prog_valid EQU 0      ;Valid Permanent programming
      414      ;occurred.
=0001      415  reset_inhibit EQU 1      ;Reset inhibit featured
      416      ; - used in validate message
=0002      417  reset_pace_trigger EQU 2  ;Reset pace trigger featured
      418      ; - used in validate message
=0003      419  pk_sense_rqst EQU 3      ;Single Peak sense measurement
      420      ;requested from programmer
=0004      421  uplnk_cnfrm   EQU 4      ;Uplink confirmation required
      422      ;on next event.
      423
      424  ;*****
      425  ;*          ULID
      426  ;*****
=0005      427
      428  CRC_error      EQU 5      ;CRC error indicator
=0006      429  uplnk_memory EQU 6      ;Uplink include memory block
=0007      430  uplnk_CRC    EQU 7      ;Uplink includes CRC and header
      431
      432  ;*****
      433  ;*          Uplink_flags
      434  ;*****
      435
      436
=0000      437  uplnk_disabled EQU 0      ;Uplink is disabled
=0001      438  uplnk_bsy     EQU 1      ;Uplink channel is busy
=0002      439  up_ram_pnd    EQU 2      ;RAM uplink pending
=0003      440  up_stat_pnd   EQU 3
=0004      441  intrrg_pnd    EQU 4      ;Interrogate data uplink pending
=0005      442  lcap_mrkr_pnd EQU 5      ;Loss of capture marker uplink
      443      ;pending
=0006      444  mrkr_pnd      EQU 6      ;Event marker uplink pending
=0007      445  meas_pnd      EQU 7      ;Measured value uplink pending
      446
=0003      447  UPLNK_GN_SET   EQU (21*uplnk_disabled + 21*uplnk_bsy)
      448      ;Disable uplink and set busy
      449      ;for gain of signal
      450  ;*****
      451  ;*          Uplink_stat equates
      452  ;*
      453  ;*****
      454
=0004      455  page0_write   EQU 4      ;Write occurred on page 0
=0005      456  magnet_applied EQU 5      ;Reed switch is closed
=0006      457  checksum_error EQU 6      ;Ram checksum error flag
=0007      458  POR_occured   EQU 7      ;POR flag
      459
=00F0      460  UPLNK_CLR_MSK   EQU 11110000B ;Clear error bits in uplink
=00C0      461      ;stat
      462  UPLNK_POR_MSK      EQU 11000000B ;Init mask used during POR
      463
      464  ;*****
      465  ;*
      466  ;*          Downlink Control Byte equates
      467  ;*

```

-16-

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 SYSTEM DATA AREA ===== File: DATA.ASM
 ===== \$Revision: 3.0 \$ =====

10/12/89 08:11:23
 Page 12

```

494 ;*****
495 ;*
496 ;*      Telemetry equates
497 ;*
498 ;*****
499
500 ;*****
501 ;*      Marker values
502 ;*****
503
=0066 504 MK_REFRAC_SENSE EQU    66H      ;Ventricular refractory sense mker
=00EE 505 MK_SENSE      EQU    0EEH      ;Ventricular sense marker
=00CC 506 MK_PACE      EQU    0CCH      ;Ventricular pace marker
=0077 507 MK_LOC      EQU    77H      ;Loss of capture marker
=00DD 508 MK_TRIGGERED EQU    0DDH      ;Triggered pace marker
509
=0080 510 UP_CRC      EQU    80H      ;Uplink CRC val for ULID regist
=0000 511 UP_NOCRC     EQU    0        ;Uplink no CRC val for ULID reg
=0040 512 UP_MEM      EQU    40H      ;Uplink mem val for ULID regist
=0000 513 UP_NOMEM    EQU    0        ;Uplink no mem val for ULID
514 ;register
515
516 ;
517 ; ID code and CRC bits for uplink messages
518 ;
=0080 519 STATUS_ID     EQU    0 + UP_CRC + UP_NOMEM ;Confirmation ID
=00C0 520 RAM_ID        EQU    0 + UP_CRC + UP_MEM   ;RAM uplink ID
=0043 521 MARKER_ID    EQU    3 + UP_NOCRC + UP_MEM ;Marker channel ID
=0044 522 PKSENSE_ID  EQU    4 + UP_NOCRC + UP_MEM ;Measure value IDs
=0046 523 PKDPDT_ID  EQU    6 + UP_NOCRC + UP_MEM
=0047 524 PKPRESS_ID  EQU    7 + UP_NOCRC + UP_MEM
=0048 525 DLTAVOLT_ID  EQU    8 + UP_NOCRC + UP_MEM
=0049 526 ACTCHT_ID  EQU    9 + UP_NOCRC + UP_MEM
=004A 527 SENSTHRS_ID EQU    10 + UP_NOCRC + UP_MEM
528
529 ;*****
530 ;*      Misc. telemetry equates
531 ;*
532 ;*****
533
=00C3 534 ACCESS_CODE EQU    0C3H      ;Telemetry access code for IPG
=00B3 535 RM_MODEL_ID EQU    10110011B ;IPG model I.D. value, model 8444
536
=0027 537 INTRRG_SIZ   EQU    39        ;Size of interrogate block
=0080 538 MAX_MEMREAD EQU    128      ;Maximum memory block read size
539
=000F 540 PG0          EQU    0FH        ;Control byte Page 0 ID
=0001 541 PG7          EQU    1         ;Control byte Page 7 ID
=0002 542 PG8          EQU    2         ;Control byte Page 8 ID
=0004 543 PG10         EQU    4        ;Control byte Page 10 ID
544
=0003 545 DNLX_EXTRA_LEN EQU    3        ;Message overhead (sub from
546 ;HW bytcount)
=0001 547 DNLX_CB_INDX  EQU    1        ;First val field in downlink
548 ;message
549 ;
550 ; Emergency values
551 ;
=0041 552 EMG_PW        EQU    41H        ;Emergency Pulse Width (2ms)
=0018 553 EMG_AMP      EQU    18H        ;Emergency pulse amplitude
554 ;(6.0 Volts)
555
=0023 556 HIGH_RATE     EQU    23H        ;Highest rate that will allow
557 ;full RAM uplink (170ppm)
=001E 558 UPLINK_DELAY EQU    1EH        ;Minimum time before next
559 ;scheduled event
560 ;needed for RAM uplink (300ms)
=0003 561 UPSTAT_DELAY  EQU    03H        ;Minimum time before next

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 EXECUTIVE ===== File: POREXEC.ASH
 ===== \$Revision: 3.0 \$ =====

10/12/89 08:12:32
 Page 8

```

264 ;*****
265 ;*          POR and Executive Macros          *
266 ;*****
267
268 ;***** CHECK_MARKER_UPLINK *****
269 ;*
270 ;* Determine which marker code to uplink while in magnet mode or
271 ;* extended telemetry. If RAM uplink is in progress, the marker
272 ;* will be ignored.
273 ;*
274 ;* ENTRY CONDITIONS:
275 ;*   A pace/sense or refractory sensed event is being processed.
276 ;*   PACESTAT indicates if the event was refractory.
277 ;*
278 ;* EXIT CONDITIONS:
279 ;*   If maker channel is active and a valid marker is detected;
280 ;*   a marker is uplinked.
281 ;*
282 ;*****
283
284 ;-----
285 ;* MACRO CHECK_MARKER_UPLINK
286 ;* BEGIN
287 ;*   (* check for marker uplink *)
288 ;*   IF (markers_active of mag_flags) THEN
289 ;*   BEGIN
290 ;-----
291 ;CHECK_MARKER_UPLINK %MACRO
292 ;CMU_START
293 ;
294 ;           ;Jump if marker channel NOT active
295 ;           BRCLR    markers_active,mag_flags,CMU_END
296 ;-----
297 ;           IF ((refractory_evnt of PACESTAT)
298 ;           AND (sensed_evnt of exec_flags)) THEN
299 ;           BEGIN   (* Refractory sensed event *)
300 ;           IF ((timeout_int - event_time) > 1) THEN
301 ;           x := MK_REFRAC_SENSE;
302 ;           ELSE
303 ;           EXIT;
304 ;           END;
305 ;-----
306 ;           ;Jump if NOT refractory sensed event
307 ;           BRCLR    refractory_evnt,PACESTAT,CMU_VVT
308 ;           BRCLR    sensed_evnt,exec_flags,CMU_VVT
309 ;           LDA      timeout_int
310 ;           SUB      event_time
311 ;           CMP      #1           ;Is there enough time for marker uplink?
312 ;           BLS      CMU_END      ; No, just exit
313 ;           LDX      MK_REFRAC_SENSE
314 ;           BRA      CMU_UL       ; Yes, load marker and go uplink it
315 ;-----
316 ;           ELSE IF ((paced_evnt of exec_flags) AND
317 ;           (sensed_evnt of exec_flags)) THEN
318 ;           BEGIN
319 ;           (* VVT mode, if triggered event send a triggered marker,
320 ;           unless output is inhibited then send sense marker. *)
321 ;           IF NOT(inhibit of tlm_flags) THEN
322 ;           x := MK_TRIGGERED;
323 ;           ELSE
324 ;           x := MK_SENSE;
325 ;           END;
326 ;-----
327 ;CMU_VVT
328 ;
329 ;           ;Jump if NOT both pace and sense
330 ;           BRCLR    paced_evnt,exec_flags,CMU_CKPACE
331 ;           BRCLR    sensed_evnt,exec_flags,CMU_CKPACE
332 ;           ;Check for output inhibited

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ***** R2 EXECUTIVE ***** File: POREXEC.ASM
 ***** \$Revision: 3.0 \$ *****

10/12/89 08:12:32
 Page 9

```

332 ;          BRCLR    inhibit_enabled,tlm_flags,CMU_INHBT
333 ;          LDX      #MK_SENSE      ;If not, get sense marker
334 ;          BRA      CMU_UL          ;Go uplink it
335 ;CMU_INHBT
336 ;          LDX      #MK_TRIGGERED ;Else get triggered marker
337 ;          BRA      CMU_UL          ;And send it
338 ;-----
339 ;;@      ELSE IF ((paced_evnt of exec_flags)
340 ;;@      AND (NOT(inhibit of tlm_flags))) THEN
341 ;;@      (* If pacing is not inhibited, send a PACE marker. *)
342 ;;@      x := MK_PACE;
343 ;-----
344 ;CMU_CKPACE
345 ;          ;Jump if NOT paced or if inhibited
346 ;          BRCLR    paced_evnt,exec_flags,CMU_CKSENSE
347 ;          BRSET    inhibit_enabled,tlm_flags,CMU_CKSENSE
348 ;          LDX      #MK_PACE      ;Else get marker code
349 ;          BRA      CMU_UL          ;And send it
350 ;-----
351 ;;@      ELSE IF (sensed_evnt of exec_flags) THEN
352 ;;@      x := MK_SENSE;
353 ;;@      ELSE
354 ;;@      (* No marker to uplink exit macro *)
355 ;;@      EXIT;
356 ;-----
357 ;CMU_CKSENSE
358 ;          ;Jump if not sensed event
359 ;          BRCLR    sensed_evnt,exec_flags,CMU_END
360 ;          LDX      #MK_SENSE      ;Else get marker value
361 ;-----
362 ;;@      (* Uplink marker code *)
363 ;;@      CALLM UPLINK_MARKER(x);
364 ;;@      END; (* marker channel active *)
365 ;-----
366 ;
367 ;CMU_UL
368 ;          UPLINK_MARKER      ;Uplink marker (value in x)
369 ;CMU_END
370 ;          XENDM
371 ;-----
372 ;@      END; (* CHECK_MARKER_UPLINK *)
373 ;-----
374 ;
375 SEJECT

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ***** R2 EXECUTIVE ***** File: POREXEC.ASH
 ***** \$Revision: 3.0 \$ *****

10/12/89 08:12:32
 Page 12

```

480 ;***** UPLINK_MARKER *****
481 ;
482 ; This macro uplinks event markers if the channel is free.
483 ;
484 ; ENTRY CONDITIONS:
485 ; This routine expects x to contain the marker value to be
486 ; uplinked.
487 ;
488 ; EXIT CONDITIONS:
489 ; If the uplink channel is available it is captured and the
490 ; marker is uplinked. If the channel is busy and there are
491 ; no pending markers the marker is flagged pending for uplink
492 ; at the end of the current uplink.
493 ;
494 ;*****
495
496 ;UPLINK_MARKER MACRO
497 ;
498 ;-----
499 ;; MACRO UPLINK_MARKER;
500 ;; BEGIN
501 ;;    disable interrupts;
502 ;;    (* Check if uplink channel is available *)
503 ;;    IF NOT(uplnk_disabled of uplink_flags) THEN
504 ;;        BEGIN
505 ;;            IF NOT(uplnk_bsy of uplink_flags) THEN
506 ;;                BEGIN
507 ;;                    (* If Uplink channel is free then uplink marker *)
508 ;;                    uplnk_bsy of uplink_flags := TRUE;
509 ;;                    enable interrupts;
510 ;;                    marker_val := x;
511 ;;                    TELADHI := HIADDR(marker_val);
512 ;;                    TELADLO := LOADDR(marker_val);
513 ;;                    BYTCOUNT := 1;
514 ;;                    ULID := MARKER_ID;
515 ;;                    RAM_uplink of TELSTAT := TRUE;
516 ;;                END;
517 ;;            END;
518 ;;        END;
519 ;;    END;
520 ;;
521 ;; UPLINK_START
522 ;;
523 ;;    SEI ;Disable interrupts
524 ;;    BRSET uplnk_disabled,uplink_flags,UPLMDONE
525 ;;
526 ;; UPLINK_MARKER
527 ;;
528 ;;    BRSET uplnk_bsy,uplink_flags,UPL_BSY
529 ;;
530 ;;    Uplink NOT busy
531 ;;
532 ;;    BSET uplnk_bsy,uplink_flags ;Flag uplink busy
533 ;;    CLI ;Enable interrupts
534 ;;    STX marker_val ;Put marker value in buffer
535 ;;    LDA #HIGH marker_val ;Get MSB of buffer address
536 ;;    STA TELADHI ;Write it to hardware
537 ;;    LDA #LOW marker_val ;Get LSB of buffer address
538 ;;    STA TELADLO ;Etc.
539 ;;
540 ;;    LDA #1 ;Get output count
541 ;;    STA BYTCOUNT ;Write to hardware count register
542 ;;
543 ;;    LDA #MARKER_ID ;Get ID code
544 ;;    STA ULID ;Tell the hardware
545 ;;    BSET RAM_uplink,TELSTAT ;Start the uplink
546 ;;    BRA UPLMDONE
547 ;;
548 ;; ELSE
549 ;; BEGIN

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ***** R2 EXECUTIVE ***** File: POREXEC.ASM
 ***** \$Revision: 3.0 \$ *****

10/12/89 08:12:32
 Page 13

```

548 ;;@          (* If no markers are pending the flag one pending *)
549 ;;@          mrkr_pnd of uplink_flags := TRUE;
550 ;;@          marker_val := x;
551 ;;@          END;
552 ;;@          END;
553 ;;@          enable interrupts;
554 ;;@
555 ;;-----
556 ;; Uplink BUSY
557 ;;
558 ;;UPL_BSY
559 ;          BSET      mrkr_pnd,uplink_flags      ;Flag marker pending and
560 ;          STX        marker_val                  ;store marker in the buffer
561 ;
562 ;;UPLMOONE
563 ;          CLI                      ;Enable interrupts
564 ;          XENDM
565 ;;-----
566 ;;@  END;      (* UPLINK_MARKER *)
567 ;;-----
568
569 SEJECT

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ***** R2 PACE OR SENSE MODULE ***** File: POS.ASM
 ***** \$Revision: 3.0 \$ *****

10/12/89 08:17:07
 Page 38

```

1816 ;***** UPLINK_INTRRG *****
1817 ;@
1818 ;@ This macro uplinks the interrogate block of size INTRRG_SIZ and
1819 ;@ starting at the address pointed to by INTRRG_AD if the uplink
1820 ;@ channel is free. Otherwise, if there is no RAM uplink, the
1821 ;@ interrogate block is set pending and is scheduled via the next
1822 ;@ TELBF interrupt, occurring when the uplink channel becomes
1823 ;@ free. All other uplinks have to be disabled while checking the
1824 ;@ uplink flags to avoid contention of the uplink channel.
1825 ;@
1826 ;@ ENTRY CONDITIONS:
1827 ;@   None.
1828 ;@
1829 ;@ EXIT CONDITIONS:
1830 ;@   None.
1831 ;@
1832 ;*****
1833
1834 ;-----
1835 ;MACRO UPLINK_INTRRG;
1836 ;BEGIN
1837 ;@ (* Capture uplink channel - If busy set interrogate pending *)
1838 ;@ disable interrupts;
1839 ;@ IF NOT(uplink_disabled of uplink_flags) THEN
1840 ;-----
1841 ;UPLINK_INTRRG MACRO
1842 ;       SEI                      ;Disable interrupts
1843 ;       BRSET    uplink_disabled,uplink_flags,UI_END
1844 ;-----
1845 ;@ BEGIN
1846 ;@ IF NOT(uplink_bsy of uplink_flags) THEN
1847 ;@ BEGIN
1848 ;@   uplink_bsy of uplink_flags := TRUE;
1849 ;@   enable interrupts;
1850 ;@   statbyt := uplink_stat;
1851 ;@   CALLM LOAD_INTRRG_UPLINK WITHIN R2L18;
1852 ;@   RAM_uplink of TELSTAT := TRUE;
1853 ;@ END;
1854 ;-----
1855 ;       BRSET    uplink_bsy,uplink_flags,UI_UBSY
1856 ;       BSET     uplink_bsy,uplink_flags
1857 ;       CLI                      ;Enable interrupts
1858 ;       LDA      uplink_stat
1859 ;       STA      statbyt         ;Initialize the uplink status byte
1860 ;UI_LIU
1861 ;       LOAD_INTRRG_UPLINK
1862 ;UI_LIU_END
1863 ;       BSET     RAM_uplink,TELSTAT
1864 ;       BRA      UI_END
1865 ;-----
1866 ;@ ELSE
1867 ;@   intrrg_pnd of uplink_flags := TRUE;
1868 ;@ END;
1869 ;-----
1870 ;UI_UBSY
1871 ;       BSET     intrrg_pnd,uplink_flags
1872 ;-----
1873 ;@ enable interrupts;
1874 ;@
1875 ;-----
1876 ;UI_END
1877 ;       CLI                      ;Enable interrupts
1878 ;       XENDM
1879 ;-----
1880 ;END; (* UPLINK_INTRRG *)
1881 ;-----
1882
1883 SEJECT

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ***** R2 LSCAP INTERRUPT MODULE ***** File: LOC.ASH
 ***** \$Revision: 3.0 \$ *****

10/12/89 08:15:30
 Page 10

```

409 ;***** UPLINK_LCAP_MARKER *****
410 ;
411 ;* This macro uplinks loss of capture markers.
412 ;*
413 ;* ENTRY CONDITIONS:
414 ;* Under magnet operations, the LSCAPINT interrupt is used for
415 ;* the uplink of LOC markers if the channel is free.
416 ;*
417 ;* EXIT CONDITIONS:
418 ;* None.
419 ;*
420 ;*****
421
422 ;-----
423 ; MACRO UPLINK_LCAP_MARKER;
424 ; BEGIN
425 ;
426 ; disable interrupts;
427 ; IF NOT (uplink_disabled of uplink_flags) THEN
428 ;-----
429 ;UPLINK_LCAP_MARKER %MACRO
430 ;ULM_START
431 ; SEI ;Disable interrupts
432 ;ULM_INT
433 ;
434 ; BRSET uplink_disabled,uplink_flags,ULM_DONE
435 ;
436 ;-----
437 ; BEGIN
438 ; IF NOT(uplink_bsy of uplink_flags) THEN
439 ;-----
440 ;
441 ; BRSET uplink_bsy,uplink_flags,ULM_LCP
442 ;
443 ;-----
444 ; BEGIN
445 ; (* If Uplink channel is free then uplink marker *)
446 ; uplink_bsy of uplink_flags := TRUE;
447 ; enable interrupts;
448 ; TELADHI := HIADDR(LCAP_MARKER);
449 ; TELADLO := LOADDR(LCAP_MARKER);
450 ; BYTCOUNT := 1;
451 ; ULID := MARKER_ID;
452 ; RAM_uplink of TELSTAT := TRUE;
453 ; END;
454 ;-----
455 ; BSET uplink_bsy,uplink_flags
456 ; CLI ;Enable interrupts
457 ; LDA #HIGH lcap_marker ;Get address MSB
458 ; STA TELADHI ;Write to controller register
459 ;
460 ; LDA #LOW lcap_marker ;Get address LSB
461 ; STA TELADLO ;Write to controller
462 ; LDA #1 ;Get byte count
463 ; STA BYTCOUNT ;Write to controller
464 ; LDA #MARKER_ID ;Get ID
465 ; STA ULID ;Write to controller
466 ; BSET RAM_uplink,TELSTAT ;Start uplink
467 ; BRA ULM_DONE ;Thats all folks
468 ;
469 ;-----
470 ; ELSE
471 ; BEGIN
472 ; (* If no markers are pending the flag one pending *)
473 ; lcap_mrkr_pnd of uplink_flags := TRUE;
474 ; END;
475 ; END;
476 ;-----

```


Avocet 6805 Assembler v2.20, #01002 Chip=146805
***** R2 LSCAP INTERRUPT MODULE ***** File: LOC.ASM
***** \$Revision: 3.0 \$ *****

10/12/89 08:15:30
Page 11

```
477 ;ULH_LCP
478 ;
479 ;           BSET      lcap_mrkr_pnd,uplink_flags      ;Jump if lcap marker pending
480 ;
481 ;-----
482 ;;a      enable interrupts;
483 ;-----
484 ;ULH_DONE
485 ;           CLI              ;Enable interrupts
486 ;
487 ;           XENDM
488 ;
489 ;-----
490 ;a
491 ;a      END;      (* UPLINK_LCAP_MARKER *)
492 ;a
493 ;-----
494 ;
495 SEJECT
```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ***** R2 ADC INTERRUPT MODULE ***** File: ADC.ASM
 ***** \$Revision: 3.0 \$ *****

10/12/89 08:14:39
 Page 6

```

187 ;*****
188 ;*                               ADC Interrupt Macros                               *
189 ;*****
190
191 ;***** UPLINK_MEAS_VAL *****
192 ;*
193 ;* This macro is used to uplink measured values.
194 ;*
195 ;* ENTRY CONDITIONS:
196 ;*   The array meas_val has been loaded with the appropriate
197 ;*   data for uplink. The number of bytes for uplink is stored
198 ;*   in the x register.
199 ;*
200 ;* EXIT CONDITIONS:
201 ;*   If the uplink channel is free it is captured and the data
202 ;*   in the meas_val buffer is uplinked. If the uplink channel
203 ;*   is busy with a RAM uplink the measured values are
204 ;*   discarded. Otherwise if the channel is busy the measured
205 ;*   values are flagged as pending and uplinked on the next
206 ;*   TELBF interrupt.
207 ;*
208 ;*****
209
210 ;-----
211 ;* MACRO UPLINK_MEAS_VAL(x);
212 ;* BEGIN
213 ;*   IF NOT(uplink_disabled of uplink_flags) THEN
214 ;-----
215 ;* UPLINK_MEAS_VAL XMACRO
216 ;* UHV_START
217 ;*       BRSET    uplink_disabled,uplink_flags,UHV_END
218 ;*
219 ;-----
220 ;* BEGIN
221 ;*   IF NOT(uplink_bsy of uplink_flags) THEN
222 ;-----
223 ;*       ;Jump if uplink busy
224 ;*       BRSET    uplink_bsy,uplink_flags,UHV_SHV
225 ;*
226 ;-----
227 ;* BEGIN
228 ;*   (* Uplink channel free uplink measured value buffer *)
229 ;*   uplink_bsy of uplink_flags := TRUE;
230 ;*   TELADHI := HIADDR(meas_val[0]);
231 ;*   TELADLO := LOADDR(meas_val[0]);
232 ;*   BYTCOUNT := x;
233 ;*   ULID := meas_id;
234 ;*   RAM_uplink of TELSTAT := TRUE;
235 ;*   END;
236 ;-----
237 ;*       BSET      uplink_bsy,uplink_flags ;Set uplink busy
238 ;*       LDA        #HIGH meas_val ;Get buffer address MSB
239 ;*       STA        TELADHI        ;Write DMA address register
240 ;*
241 ;*       LDA        #LOW meas_val  ;Get buffer address LSB
242 ;*       STA        TELADLO        ;etc.
243 ;*       STX        BYTCOUNT       ;Write byte count
244 ;*       LDA        meas_id        ;Get ID
245 ;*       STA        ULID           ;Write to hardware
246 ;*       BSET      RAM_uplink,TELSTAT ;Start uplink
247 ;*       BRA        UHV_END        ;Go exit
248 ;*
249 ;-----
250 ;* ELSE (* NOT uplink_bsy *)
251 ;* BEGIN
252 ;*   (* Set measured value uplink pending *)
253 ;*   meas_count := x;
254 ;*   meas_pnd of uplink_flags := TRUE;
255 ;*   END;
256 ;*   END;
257 ;-----
258 ;*

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
***** R2 ADC INTERRUPT MODULE ***** File: ADC.ASM
***** \$Revision: 3.0 \$ *****

10/12/89 08:14:39
Page 7

```
259 ;UMV_SMV
260 ;          STX      meas_count      ;Save pending byte count
261 ;          BSET     meas_pnd,uplink_flags ;Show pending upling
262 ;          BRA      UMV_END          ;Thats all folks
263 ;UMV_END
264 ;          XENDM
265 ;-----
266 ;@  END;      (* UPLINK_MEAS_VAL *)
267 ;-----
268
269 SEJECT
```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 TELEMETRY MODULE ===== File: TLM.ASH
 ===== \$Revision: 3.3 \$ =====

1/18/90 10:35:56
 Page 7

```

378 ;***** TLM *****
379 ;a*
380 ;a* R2, Pacemaker Model 8444
381 ;a* MODULE: TLM
382 ;a*
383 ;a* The TLM module processes magnet mode operations while the reed
384 ;a* switch is closed. These include the handling of the telemetry
385 ;a* protocol, the TMT and lead test activation, the pulse pressure
386 ;a* calculation for loss of capture markers detection. The
387 ;a* telemetry protocol involves processing downlink and uplink
388 ;a* messages. Downlink messages are validated before being acted
389 ;a* upon. The uplink consists of confirmation and confirmation +
390 ;a* replies to downlink requests.
391 ;a*
392 ;a* Routines defined in this module include:
393 ;a*
394 ;a* Macros:
395 ;a* DO_MEMWRITE - transfer downlink record to
396 ;a* memory
397 ;a* EXEC_SPEC_FUNC - decode and execute special
398 ;a* function
399 ;a* EXEC_SPEC_REQ - decode and execute special
400 ;a* requests
401 ;a* PROCESS_MEMWRITE - transfer downlink record to
402 ;a* memory and evaluate it
403 ;a* PROCESS_MSG - decode memory offsets
404 ;a* SWITCH_TO_NON_MAGMODE - restore non_magnet mode
405 ;a* operation
406 ;a* VALIDATE_MSG - validate downlink message
407 ;a*
408 ;a* Procedures:
409 ;a* None.
410 ;a*
411 ;a* Drivers:
412 ;a* GNLSINT_PROC - gain or loss interrupt handler
413 ;a* RDSWINT_PROC - reed-switch interrupt handler
414 ;a* TELBFINT_PROC - telemetry buffer interrupt
415 ;a* handler
416 ;a*
417 ;*****
418
419
420 DEFSEG TLM, CLASS=CODE
421 SEG TLM
422
423 $SETLN(MACROS.INC); %INCLUDE "MACROS.INC"

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
***** R2 TELEMETRY MODULE ***** File: TLM.ASM
***** \$Revision: 3.3 \$ *****

1/18/90 10:35:56
Page 8

```
424
425 $NOALLPUBLIC
426 $SETLN(EQUATES.INC);      XINCLUDE "EQUATES.INC"
```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
===== R2 TELEMETRY MODULE ===== File: TLM.ASH
===== \$Revision: 3.3 \$ =====

1/18/90 10:35:56
Page 33

```
1812 ;*****  
1813 ;* Telemetry Subroutines *  
1814 ;*****  
1815 ;  
1816 SEJECT
```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 TELEMETRY MODULE ===== File: TLH.ASH
 ===== \$Revision: 3.3 \$ =====

1/18/90 10:35:56
 Page 34

```

1817 ;*****
1818 ;*          Telemetry Interrupt Handlers          *
1819 ;*****
1820 ;a***** GNLSINT_PROC *****
1821 ;a*
1822 ;a* This procedure is the gain/loss interrupt handler and it is *
1823 ;a* non-preemptive. It is responsible for controlling the downlink *
1824 ;a* and disabling uplink. Whether the interrupt is do to a gain or a *
1825 ;a* loss of signal can be determined by reading a bit in the TELSTAT *
1826 ;a* register. At the beginning of a downlink all pending uplinks are *
1827 ;a* abandoned and the TELBF interrupt is masked out until the end of *
1828 ;a* downlink. In which case it is reenabled, after being first *
1829 ;a* cleared, in the case of downlink overrun. Downlink is then *
1830 ;a* disabled until just before the uplink response, either a status *
1831 ;a* uplink or a RAM uplink.
1832 ;a*
1833 ;a* ENTRY CONDITIONS:
1834 ;a*   No other interrupts are enabled at this point, ADC interrupts *
1835 ;a*   are the only higher priority and they are ignored during *
1836 ;a*   telemetry.
1837 ;a*
1838 ;a* EXIT CONDITIONS:
1839 ;a*   None.
1840 ;a*
1841 ;a*****
1842
1843 ;-----
1844 ;aPROCEDURE GNLSINT_PROC;
1845 ;aBEGIN
1846 ;a
1847 ;a (* Check if gain or loss of signal occurred. *)
1848 ;a IF (downlnk_present of TELSTAT) THEN
1849 ;a   BEGIN
1850 ;a     (* Gain of downlink signal. Clear pending uplinks, disable
1851 ;a       uplink and TELBF interrupts, and clear any ADC and
1852 ;a       TELBFINT interrupts. *)
1853 ;a     uplink_flags := UPLNK_GN_SET;
1854 ;a     IF (TMT of mag_flags) THEN
1855 ;a       reset_TMT of mag_flags := TRUE;
1856 ;a       TELBFINT of ipgstate_msk := TRUE;
1857 ;a       IRQREG := TELBFINT_MSK;
1858 ;a       ULID := 0;
1859 ;a
1860 ;a     (* If POS currently executing then postpone loss-of-signal
1861 ;a       processing until after POS is complete. *)
1862 ;a     IF ((sensed_evnt of exec_flags)
1863 ;a       OR (paced_evnt of exec_flags)) THEN
1864 ;a       GNLSINT of current_pri := TRUE;
1865 ;a   END;
1866 ;a
1867 ;-----
1868 GNLSINT_PROC
1869          BRCLR    downlnk_present,TELSTAT,GNLS_LOSS
1870          LDA      #UPLNK_GN_SET
1871          STA      uplink_flags      ;Disable uplink
1872          BRCLR    TMT,mag_flags,GNLS_NTMT
1873          BSET     reset_TMT,mag_flags ;Reset TMT if active
1874 GNLS_NTMT
1875          BSET     TELBFINT,ipgstate_msk ;Mask TELBF interrupts
1876          LDX      #TELBFIN_MSK
1877          STX      IRQREG             ;Clear TELBF interrupts
1878          CLRA
1879          STA      ULID               ;Clear ULID register
1880          LDA      exec_flags         ;Is POS currently executing?
1881          AND      #((1 SHL sensed_evnt) + (1 SHL paced_evnt))
1882          BEQ      GNLS_NPOS          ; No, then exit
1883          BSET     GNLSINT,current_pri ; Yes, postpone loss-o-signal
1884                                     ; until after POS

```

0000& 09 00* 1D
 0003& A6 03
 0005& B7 00*
 0007& 09 00* 02
 000A& 1A 00*

000C& 14 00*
 000E& AE 04
 0010& 8F 00*
 0012& 4F
 0013& B7 00*
 0015& B6 00*
 0017& A4 03
 0019& 27 02
 0018& 12 00*

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 TELEMETRY MODULE ===== File: TLM.ASH
 ===== \$Revision: 3.3 \$ =====

1/18/90 10:35:56
 Page 35

```

001D& CC 0384&      1885 GNLS_NPOS
                      1886 JHP GNLS_DONE
                      1887 ;-----
                      1888 ;@ ELSE IF (uplnk_disabled of uplnk_flags) THEN
                      1889 ;@ BEGIN
                      1890 ;@ (* If the uplnk_disabled bit was not set then a downlink
                      1891 ;@ overrun has occurred (gain of signal was missed) and
                      1892 ;@ downlink should be ignored!! *)
                      1893 ;@
                      1894 ;@ IF (reset_TMT of mag_flags) THEN
                      1895 ;@ CALL TMT_RESET WITHIN R2LIB;
                      1896 ;@
                      1897 ;@ uplnk_stat := (uplnk_stat AND UPLNK_CLR_MSK);
                      1898 ;@
                      1899 ;@ IF (TELBFININT of IRQREG) THEN
                      1900 ;@ BEGIN
                      1901 ;@ (* Downlink overflow - Flag error, uplink status,
                      1902 ;@ and clear TELBF interrupt *)
                      1903 ;@ IRQREG := TELBFINT_MSK;
                      1904 ;@ up_stat_pnd of uplnk_flags := TRUE;
                      1905 ;@ uplnk_stat := uplnk_stat OR DNK_OVF_ERR;
                      1906 ;@ END;
                      1907 ;-----
0020& 00 00* 03      1908 GNLS_LOSS
0023& CC 0384&      1909 BRSET uplnk_disabled,uplnk_flags,GNLS_LCONT
                      1910 JHP GNLS_DONE
                      1911 GNLS_LCONT
0026& 08 00* 03      1912 BRCLR reset_TMT,mag_flags,GNLS_NTHMTRST
0029& CD 0000*      1913 JSR TMT_RESET ;Go abort TMT sequence
                      1914 GNLS_NTHMTRST
002C& B6 00*      1915 LDA uplnk_stat
002E& A4 F0      1916 AND #UPLNK_CLR_MSK
0030& B7 00*      1917 STA uplnk_stat ;Mask error bits in uplink status
0032& 05 00* 0D      1918 BRCLR TELBFINT,IRQREG,GNLS_NOVF ;Has downlink overflow occurred?
0035& AE 04      1919 LDX #TELBFININT_MSK
0037& BF 00*      1920 STX IRQREG ;Clear TELBF interrupts
0039& 16 00*      1921 BSET up_stat_pnd,uplnk_flags ;Set status uplink pending
003B& AA 09      1922 ORA #DNK_OVF_ERR
003D& B7 00*      1923 STA uplnk_stat ;Set and store Overflow error
003F& CC 0350&      1924 JHP GNLS_UPLNK
                      1925 ;-----
                      1926 ;@ ELSE
                      1927 ;@ BEGIN
                      1928 ;@ (* No downlink overflow *)
                      1929 ;@ CALLM VALIDATE_MSG;
                      1930 ;@ END;
                      1931 ;-----
0020& 00 00* 03      1932 GNLS_NOVF
0023& CC 0384&      1933 ;VALIDATE_MSG
                      1934 ;-----
                      1935 ;@ (* Request event time to be latched (write any value)
                      1936 ;@ NOTE: event time takes 0.244msec to be latched *)
                      1937 ;@ EVENTIME := 0;
                      1938 ;@
                      1939 ;@ IF ((up_RAM_pnd of uplnk_flags)
                      1940 ;@ OR (intrrg_pnd of uplnk_flags)) THEN
                      1941 ;@ BEGIN
                      1942 ;@ (* Only allow RAM uplink if the pacing interval is above
                      1943 ;@ HIGH_RATE, otherwise clear uplink status flag. *)
                      1944 ;@ IF (timeout_int < HIGH_RATE) THEN
                      1945 ;@ BEGIN
                      1946 ;@ up_RAM_pnd of uplnk_flags := FALSE;
                      1947 ;@ intrrg_pnd of uplnk_flags := FALSE;
                      1948 ;@ up_stat_pnd of uplnk_flags := TRUE;
                      1949 ;@ END;
                      1950 ;@ ELSE
                      1951 ;@ up_stat_pnd of uplnk_flags := FALSE;
                      1952 ;@ END;

```


Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 TELEMETRY MODULE ===== File: TLM.ASM
 ===== \$Revision: 3.3 \$ =====

1/18/90 10:35:56
 Page 36

```

1953 ;-----
1954 GNLS_UPLNK
1955 STA EVENTIME ;Latch event time count
1956 GNLS_UPEVNT
1957 LDA uplink_flags
1958 AND #((1 SHL up_RAM_pnd) + (1 SHL intrrg_pnd))
1959 BEQ GNLS_NRAMUP ;Jump if no RAM of interrogate uplink
1960 LDA timeout_int
1961 CMP #HIGH_RATE ;Is timeout less then upper rate limit?
1962 BHS GNLS_RTLO ; No, set uplink status flag false
1963 BCLR up_RAM_pnd,uplink_flags
1964 BCLR intrrg_pnd,uplink_flags
1965 BSET up_stat_pnd,uplink_flags
1966 BRA GNLS_NRAMUP
1967 GNLS_RTLO
1968 BCLR up_stat_pnd,uplink_flags
1969 GNLS_NRAMUP
1970 ;-----
1971 ;a (* If IPG in VVT mode switch to VVI mode until next event
1972 ;a and scedule uplink if there is enough time. *)
1973 ;a triggered_mode of PACEMODE := FALSE;
1974 ;a a := timeout_int - EVENTIME
1975 ;a IF (((a > UPSTAT_DELAY) AND (up_stat_pnd of uplink_flags))
1976 ;a OR (a > UPLINK_DELAY)) THEN
1977 ;a CALL SCHEDULE_UPLINK WITHIN R2L18;
1978 ;a ELSE
1979 ;a uplnk_cnfrm of tlm2_flags := TRUE;
1980 ;a
1981 ;-----
1982 BCLR triggered_mode,PACEMODE ;Set in non-VVT mode
1983 LDA timeout_int
1984 SUB EVENTIME ;Determine time remaining before next event
1985 CMP #UPLINK_DELAY ;Enough time for block uplink?
1986 BHI GNLS_SU ; Yes, then schedule uplink
1987 BRCLR up_stat_pnd,uplink_flags,GNLS_NUPLNK
1988 CMP #UPSTAT_DELAY ;Enough time for status uplink?
1989 BLS GNLS_NUPLNK ; No, don't attempt uplink
1990 GNLS_SU
1991 JSR SCHEDULE_UPLINK
1992 BRA GNLS_CTLBF
1993 GNLS_NUPLNK
1994 BSET uplnk_cnfrm,tlm2_flags ;Indicate uplink to follow next event
1995 ;-----
1996 ;a (* Enable TELBF interrupts and clear ADC interrupts *)
1997 ;a TELBFINT of ipgstate_msk := FALSE;
1998 ;a END;
1999 ;a
2000 ;a IRQREG := ADCINT_MSK;
2001 ;-----
2002 GNLS_CTLBF
2003 BCLR TELBFINT,ipgstate_msk
2004 GNLS_DONE
2005 LDA #ADCINT_MSK
2006 STA IRQREG ;Clear pending ADC interrupts
2007 GNLS_END
2008 RTS
2009 ;-----
2010 ;aEND; (* GNLSINT_PROC *)
2011 ;-----
2012
2013 SEJECT

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 TELEMETRY MODULE ===== File: TLH.ASM
 ===== \$Revision: 3.3 \$ =====

1/18/90 10:35:56
 Page 38

```

2071 ;***** TELBFINT_PROC *****
2072 ;*
2073 ;* This procedure is the telemetry buffer interrupt handler. It is *
2074 ;* non-preemptive. It is responsible for scheduling pending uplinks *
2075 ;* (i.e. markers). If the last uplink was a RAM uplink, all pending *
2076 ;* uplinks are cancelled. Otherwise, if there is either a pending *
2077 ;* interrogate block or measured value, they are uplinked. *
2078 ;*
2079 ;* ENTRY CONDITIONS:
2080 ;* No other interrupts are allowed during this routine, ADC must *
2081 ;* be cleared if one occurred during uplink reschedule, and *
2082 ;* processing of GAIN/LOSS must wait until after uplink TELBF *
2083 ;* completes to insure that the uplink flags are not corrupted *
2084 ;*
2085 ;* EXIT CONDITIONS:
2086 ;* None.
2087 ;*
2088 ;*****
2089
2090 ;-----
2091 ;PROCEDURE TELBFINT_PROC;
2092 ;BEGIN
2093 ;
2094 ; (* If RAM uplink complete clear all pending uplinks *)
2095 ; IF (uplink_disabled of uplink_flags) THEN
2096 ;   uplink_flags := 0;
2097 ;
2098 TELBFINT_PROC
2099       BRCLR    uplink_disabled,uplink_flags,TLBF_UPLNK
2100       CLRA
2101       STA      uplink_flags    ;Clear all pending uplinks
2102       BRA      TLBF_DONE
2103 ;-----
2104 ; ELSE
2105 ; BEGIN
2106 ;   (* Previous uplink was not a RAM uplink, uplink pending *)
2107 ;   IF (mrkr_pnd of uplink_flags) THEN
2108 ;     BEGIN
2109 ;       (* Marker from POS is pending *)
2110 ;       mrkr_pnd of uplink_flags := FALSE;
2111 ;       TELADHI := HIADDR(marker_val[0]);
2112 ;       TELADLO := LOADDR(marker_val[0]);
2113 ;       BYTCOUNT := marker_cnt;
2114 ;       ULID := MARKER_ID;
2115 ;       RAM_uplink of TELSTAT := TRUE;
2116 ;       END;
2117 ;
2118 ;-----
2119 TELBF_UPLNK
2120       BRCLR    mrkr_pnd,uplink_flags,TLBF_LCAP
2121       BCLR     mrkr_pnd,uplink_flags
2122       LDA      #HIGH marker_val    ;Load register with hi address of marker value
2123                                     address
2124       STA      TELADHI
2125       LDA      #LOW marker_val     ;Load register with low address of marker value
2126                                     address
2127       STA      TELADLO
2128       LDX      marker_cnt          ;Load x with byte count
2129       LDA      #MARKER_ID          ;Load a with marker identification byte
2130       BRA      TLBF_STRTU
2131 ;-----
2132 ; ELSE IF (lcap_mrkr_pnd of uplink_flags) THEN
2133 ; BEGIN
2134 ;   (* Marker from loss of capture is pending *)
2135 ;   lcap_mrkr_pnd of uplink_flags := FALSE;
2136 ;   TELADHI := HIADDR(lcap_marker);
2137 ;   TELADLO := LOADDR(lcap_marker);
2138 ;   BYTCOUNT := 1;

```

0423& 01 00* 05
 0426& 4F
 0427& 87 00*
 0429& 20 68

0428& 0D 00* 11
 042E& 1D 00*
 0430& A6 ...X

0432& 87 00*
 0434& A6 ...X

0436& 87 00*
 0438& CE 0000*
 043B& A6 43
 043D& 20 49

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 TELEMETRY MODULE ===== File: TLM.ASH
 ===== \$Revision: 3.3 \$ =====

1/18/90 10:35:56
 Page 39

```

2137 ;@      ULID := MARKER_ID;
2138 ;@      RAM_uplink of TELSTAT := TRUE;
2139 ;@      END;
2140 ;@
2141 ;-----
2142 TLBF_LCAP
043F& 08 00* 10 2143      BRCLR    lcap_mkr_pnd,uplink_flags,TLBF_INTRRG
0442& 18 00* 2144      BCLR     lcap_mkr_pnd,uplink_flags
0444& A6 ..X 2145      LDA      #HIGH lcap_marker ;Load register with hi address of lcap marker v
                                alue address
0446& B7 00* 2146      STA      TELADHI
0448& A6 ..X 2147      LDA      #LOW lcap_marker ;Load register with low address byte of lcap ma
                                rker value address
044A& B7 00* 2148      STA      TELADLO
044C& AE 01 2149      LD      #1 ;Load x with byte count
044E& A6 43 2150      LDA      #MARKER_ID ;Load a with marker identification byte
0450& 20 36 2151      BRA      TLBF_STRTU
2152 ;-----
2153 ;@      ELSE IF (intrrg_pnd of uplink_flags) THEN
2154 ;@      BEGIN
2155 ;@      intrrg_pnd of uplink_flags := FALSE;
2156 ;@      statbyt := uplink_stat;
2157 ;@      CALLM LOAD_INTRRG_UPLINK WITHIN R2LIB;
2158 ;@      RAM_uplink of TELSTAT := TRUE;
2159 ;@      END;
2160 ;-----
2161 TLBF_INTRRG
0452& 09 00* 20 2162      BRCLR    intrrg_pnd,uplink_flags,TLBF_MEAS
0455& 19 00* 2163      BCLR     intrrg_pnd,uplink_flags
0457& B6 00* 2164      LDA      uplink_stat
0459& C7 0000* 2165      STA      statbyt ;Update status byte
2166 TLBF_LDIN
2167 ;LOAD_INTRRG_UPLINK
2168 TLBF_LDIN_END
0471& 16 00* 2169      BSET     RAM_uplink,TELSTAT ;Initiate uplink
0473& 20 1E 2170      BRA      TLBF_DONE
2171 ;-----
2172 ;@      ELSE IF (meas_pnd of uplink_flags) THEN
2173 ;@      BEGIN
2174 ;@      meas_pnd of uplink_flags := FALSE;
2175 ;@      TELADHI := HIADDR(meas_val[0]);
2176 ;@      TELADLO := LOADDR(meas_val[0]);
2177 ;@      BYTCOUNT := meas_count;
2178 ;@      ULID := meas_id;
2179 ;@      RAM_uplink of TELSTAT := TRUE;
2180 ;@      END;
2181 ;-----
2182 TLBF_MEAS
0475& 0F 00* 18 2183      BRCLR    meas_pnd,uplink_flags,TLBF_NUPLNK
0478& 1F 00* 2184      BCLR     meas_pnd,uplink_flags
047A& A6 ..X 2185      LDA      #HIGH meas_val ;Load register with hi address of measured valu
                                e address
047C& B7 00* 2186      STA      TELADHI
047E& A6 ..X 2187      LDA      #LOW meas_val ;Load register with low address byte of measure
                                d value address
0480& B7 00* 2188      STA      TELADLO
0482& CE 0000* 2189      LD      meas_count ;Load x with byte count
0485& C6 0000* 2190      LDA      meas_id ;Load a with marker identification byte
2191 TLBF_STRTU
2192      STX      BYTCOUNT ;Store byte count
2193      STA      ULID ;Store marker identification byte
2194      BSET     RAM_uplink,TELSTAT ;Set the telemetry status byte and exit
2195      BRA      TLBF_DONE
2196 ;-----
2197 ;@      ELSE (* No pending uplinks *)
2198 ;@      uplink_flags := 0;
2199 ;@      END;
2200 ;@      (* Clear pending ADC interrupts *)

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
===== R2 TELEMETRY MODULE ===== File: TLM.ASM
===== \$Revision: 3.3 \$ =====

1/18/90 10:35:56
Page 40

```
0490& 4F      2201  ;@ IRQREG := ADCINT_MSK;
0491& B7 00*   2202  ;@
                2203  ;-----
                2204  TLBF_NUPLNK
                2205          CLRA
0493& A6 01     2206          STA      uplink_flags  ;Clear uplink flags, no uplinks pending
0495& B7 00*   2207  TLBF_DONE
                2208          LDA      #ADCINT_MSK
                2209          STA      IRQREG      ;Clear pending ADC interrupts
0497& 81       2210  TLBF_END
                2211          RTS
                2212  ;-----
                2213  ;@END;  (* TELBFINT_PROC *)
                2214  ;@
                2215  ;-----
                2216          END
```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 LIBRARY MODULE ===== File: R2LIB.ASM
 ===== \$Revision: 3.3 \$ =====

1/08/90 11:38:20
 Page 7

```

120 ;***** LOAD_INTRRG_UPLINK *****
121 ;*
122 ;* This macro loads the telemetry registers in preparation for an
123 ;* Interrogate block uplink.
124 ;*
125 ;* ENTRY CONDITIONS:
126 ;* Uplink data registers are ready to be loaded without conflict.
127 ;*
128 ;* EXIT CONDITIONS:
129 ;* The interrogate block the size of INTRRG_SIZ and starting at
130 ;* the address pointed to by INTRRG_AD is setup for uplink.
131 ;*
132 ;*****
133
134 ;-----
135 ;a MACRO LOAD_INTRRG_UPLINK;
136 ;a BEGIN
137 ;a
138 ;a (* Load interrogate status byte *)
139 ;a intrrg_r2_stat := R2_stat;
140 ;a (* Uplink channel assumed free and uplink_disabled bit set *)
141 ;a TELADHI := HI BYTE(INTRRG_AD);
142 ;a TELADLO := LO BYTE(INTRRG_AD);
143 ;a BYTCOUNT := INTRRG_SIZ;
144 ;a ULID := RAM_ID;
145 ;a
146 ;a END; (* LOAD_INTRRG_UPLINK *)
147 ;-----
148 ;LOAD_INTRRG_UPLINK xMACRO
149 ; LDA r2_stat ;Get r2 status byte
150 ; STA intrrg_r2_stat ;put in interrogate status byte
151 ; LDA #HIGH INTRRG_AD ;Get address hi byte
152 ; STA TELADHI ;Send it to the hardware
153 ;
154 ; LDA #LOW INTRRG_AD ;Get address lo byte
155 ; STA TELADLO ;Send it to the hardware
156 ;
157 ; LDA #INTRRG_SIZ ;Get byte count
158 ; STA BYTCOUNT ;Write hardware register
159 ; LDA #RAM_ID ;Get ID
160 ; STA ULID ; etc. etc. etc.
161 ; XENDM
162
163 SEJECT

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 LIBRARY MODULE ===== File: R2LIB.ASM
 ===== SRevision: 3.3 \$ =====

1/08/90 11:38:20
 Page 8

```

164 ;***** LOAD_RAM_UPLINK *****
165 ;*
166 ;* This macro loads the telemetry registers in preparation for a
167 ;* RAM block uplink.
168 ;*
169 ;* ENTRY CONDITIONS:
170 ;* Uplink data registers are ready to be loaded without conflict.
171 ;*
172 ;* EXIT CONDITIONS:
173 ;* A RAM block of length indicated by P_rd_bytes starting at the
174 ;* address indicated by P_rd_start is setup for uplink.
175 ;*
176 ;*****
177
178 ;-----
179 ; MACRO LOAD_RAM_UPLINK;
180 ; BEGIN
181 ;
182 ; (* Uplink channel assumed free and uplink_disabled bit set *)
183 ; intrrg_r2_stat := R2_stat;
184 ; TELADHI := HI BYTE(P_rd_start);
185 ; TELADLO := LOW BYTE(P_rd_start);
186 ; BYTCOUNT := P_rd_bytes;
187 ; ULID := RAM_ID;
188 ;
189 ; END; (* LOAD_RAM_UPLINK *)
190 ;-----
191 ;LOAD_RAM_UPLINK %MACRO
192 ; LDA r2_stat ;Get r2 status byte
193 ; STA intrrg_r2_stat ;put in interrogate status byte
194 ; LDA P_rd_start ;Get address hi byte
195 ; STA TELADHI ;Send it to the hardware
196 ;
197 ; LDA P_rd_start +1 ;Get address lo byte
198 ; STA TELADLO ;Send it to the hardware
199 ;
200 ; LDA P_rd_bytes ;Get byte count
201 ; STA BYTCOUNT ;Write hardware register
202 ; LDA #RAM_ID ;Get ID
203 ; STA ULID ; etc. etc. etc.
204 ; XENDM
205
206
207 $RESETLN
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240 $NOALLPUBLIC
241 $NOLIST ;Don't List the equate file

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 LIBRARY MODULE ===== File: R2LIB.ASM
 ===== \$Revision: 3.3 \$ =====

1/08/90 11:38:20
 Page 29

```

1106 ;***** SCHEDULE_UPLINK *****
1107 ;*
1108 ;* This procedure schedules uplink of RAM, Interrogate block, or
1109 ;* status in this order of priority.
1110 ;*
1111 ;* ENTRY CONDITIONS:
1112 ;* No other interrupts are allowed during this routine, ADC
1113 ;* interrupts must be cleared if one occurred during uplink
1114 ;* scheduling. Processing of the GAIN/LOSS and TLBF interrupts
1115 ;* wait until after uplink is scheduled to ensure that the
1116 ;* uplink flags are not corrupted.
1117 ;*
1118 ;* EXIT CONDITIONS:
1119 ;* Either a RAM block, an Interrogate block, or a status
1120 ;* confirmation block are uplinked if any are pending.
1121 ;* Status is imbedded in a RAM or Interrogate block uplink.
1122 ;*
1123 ;*****
1124
1125 ;-----
1126 ;a PROCEDURE SCHEDULE_UPLINK;
1127 ;a BEGIN
1128 ;a
1129 ;a (* Load status byte for RAM uplink and the load telemetry
1130 ;a registers for uplink. *)
1131 ;a IF (up_RAM_pnd of uplink_flags) THEN
1132 ;a BEGIN
1133 ;a (* Load for Ram uplink *)
1134 ;a CALLM LOAD_RAM_UPLINK;
1135 ;a up_RAM_pnd of uplink_flags := FALSE;
1136 ;a END;
1137 ;-----
1138 SCHEDULE_UPLINK
1139 ;Jump if NOT RAM uplink
1140 BRCLR up_RAM_pnd,uplink_flags,SUP_INTRRG
1141
1142 SU_LRU
1143 ;LOAD_RAM_UPLINK
1144 LDA r2_stat ;Get r2 status byte
1145 STA Intrrg_r2_stat ;put in interrogate status byte
1146 LDA P_rd_start ;Get address hi byte
1147 STA TELADHI ;Send it to the hardware
1148 LDA P_rd_start +1 ;Get address lo byte
1149 STA TELADLO ;Send it to the hardware
1150
1151 LDA P_rd_bytes ;Get byte count
1152 STA BYTCOUNT ;Write hardware register
1153 LDA #RAM_ID ;Get ID
1154 STA ULID ; etc. etc. etc.
1155
1156 SU_LRU_END
1157 BCLR up_RAM_pnd,uplink_flags ;Clear the pending flag
1158 JMP SUP_STRT ;Go start uplink
1159
1160 ;-----
1161 ;a ELSE IF (Intrrg_pnd of uplink_flags) THEN
1162 ;a BEGIN
1163 ;a (* Load for Interrogate block uplink *)
1164 ;a CALLM LOAD_INTRRG_UPLINK WITHIN R2LIB;
1165 ;a Intrrg_pnd of uplink_flags := FALSE;
1166 ;a END;
1167 ;-----
1168 SUP_INTRRG
1169 ;Jump if NOT Interrogate
1170 BRCLR Intrrg_pnd,uplink_flags,SUP_STAT
1171
1172 SU_LIU
1173 ;LOAD_INTRRG_UPLINK
1174 LDA r2_stat ;Get r2 status byte
1175 STA Intrrg_r2_stat ;put in interrogate status byte

```

Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ===== R2 LIBRARY MODULE ===== File: R2LIB.ASM
 ===== \$Revision: 3.3 \$ =====

1/08/90 11:38:20
 Page 30

```

01A8& A6 ..X      1174          LDA      #HIGH INTRRG_AD ;Get address hi byte
01AA& B7 00*      1175          STA      TELADHI      ;Send it to the hardware
                  1176
01AC& A6 ..X      1177          LDA      #LOW INTRRG_AD ;Get address lo byte
01AE& B7 00*      1178          STA      TELADLO      ;Send it to the hardware
                  1179
01B0& A6 00*      1180          LDA      #INTRRG_SIZ  ;Get byte count
01B2& B7 00*      1181          STA      BYTCOUNT    ;Write hardware register
01B4& A6 C0       1182          LDA      #RAM_ID    ;Get ID
01B6& B7 00*      1183          STA      ULID      ; etc. etc. etc.
                  1184
01B8& 19 00*      1185          BCLR     intrrg_pnd,uplink_flags ;Clear the flag
01BA& CC 01D1&    1186          JMP      SUP_STRT    ;Go start uplink
                  1187
                  1188          ;-----
                  1189          ;& ELSE IF (up_stat_pnd of uplink_flags) THEN
                  1190          ;& BEGIN
                  1191          ;& (* Load for status ID byte for uplink *)
                  1192          ;& ULID := STATUS_ID;
                  1193          ;& up_stat_pnd of uplink_flags := FALSE;
                  1194          ;& END;
                  1195          ;-----
1196          SUP_STAT
1197          ;& Jump if NOT status ID byte
1198          BRCLR    up_stat_pnd,uplink_flags,SUP_NO_UP
1199          LDA      #STATUS_ID
1200          STA      ULID      ;Write status ID to hardware
1201          BCLR     up_stat_pnd,uplink_flags ;Clear the flag
1202          BRA      SUP_STRT    ;Go start Uplink
1203
1204          ;-----
1205          ;& ELSE
1206          ;& BEGIN
1207          ;& (* No uplink scheduled reset telemetry and exit routine *)
1208          ;& uplink_flags := 0;
1209          ;& CALL SET_TLM_TYPE
1210          ;& downlink_enable of TELSTAT := TRUE;
1211          ;& EXIT;
1212          ;& END;
1213          ;-----
1214          SUP_NO_UP
1215          CLR      uplink_flags ;Clear uplink_flags, no uplink
1216          JSR      SET_TLM_TYPE ;Set telemetry type and enable downlink
1217          BSET     downlink_enabled,TELSTAT
1218          BRA      SUP_END      ;Go exit
1219
1220          ;-----
1221          ;& (* Set telemetry type start uplink and enable downlink *)
1222          ;& statbyt := uplink_stat;
1223          ;& CALL SET_TLM_TYPE;
1224          ;& downlink_enable of TELSTAT := TRUE;
1225          ;& RAM_uplink of TELSTAT := TRUE;
1226          ;&
1227          ;-----
1228          SUP_STRT
1229          LDA      uplink_stat ;Get uplink status
1230          STA      statbyt
1231          JSR      SET_TLM_TYPE ;Set telemetry type
1232          ORA      #((1 SHL RAM_uplink) + (1 SHL downlink_enabled))
1233          STA      TELSTAT    ;enable downlink and start uplink
1234          SUP_END
1235          RTS              ;Return to caller
1236          ;-----
1237          ;& END; (* SCHEDULE_UPLINK *)
1238          ;-----
1239
1240          SEJECT

```


Avocet 6805 Assembler v2.20, #01002 Chip=146805
 ***** R2 LIBRARY MODULE ***** File: R2LIB.ASM
 ***** \$Revision: 3.3 \$ *****

1/08/90 11:38:20
 Page 31

```

1241 ;***** SET_TLM_TYPE *****
1242 ;a
1243 ;a This procedure decodes the telemetry type in P_tlm_type and
1244 ;a sets up the hardware and marker channel accordingly.
1245 ;a
1246 ;a ENTRY COND. :
1247 ;a P_tlm_type contains the desired telemetry.
1248 ;a
1249 ;a EXIT COND. :
1250 ;a The analog uplink telemetry is updated on the next frame.
1251 ;a Curr_tlm_type is written to PACESTAT and may not equal
1252 ;a P_tlm_type.
1253 ;a a - contains the current value of the TELSTAT register.
1254 ;a
1255 ;*****
1256
1257 ;-----
1258 ;a PROCEDURE SET_TLM_TYPE;
1259 ;a BEGIN
1260 ;a
1261 ;a (* test for markers uplink selected *)
1262 ;a IF (marker_enabled of P_tlm_type := TRUE) THEN
1263 ;a   marker_active of mag_flags := TRUE;
1264 ;a ELSE
1265 ;a   marker_active of mag_flags := FALSE;
1266 ;a
1267 ;-----
1268 SET_TLM_TYPE
1269     LDA     P_tlm_type    ;Jump if idle markers set
1270     AND     #1 SHL marker_enabled)
1271     BEQ     STT_ICLR
1272     BSET     markers_active,mag_flags ;Show idle markers
1273     BRA     STT_ADJ      ;Go adjust telem type
1274 STT_ICLR
1275     BCLR     markers_active,mag_flags
1276
1277 ;-----
1278 ;a (* adjust the telemetry type *)
1279 ;a curr_tlm_type := (P_tlm_type AND TLM_TYPE_MSK) OR IDLE_UPLINK;
1280 ;a TELSTAT := (TELSTAT AND TELSTAT_MSK) OR curr_tlm_type;
1281 ;a
1282 ;-----
1283 STT_ADJ
1284     LDA     P_tlm_type    ;Get telemetry type
1285     AND     #TLM_TYPE_MSK ;Isolate real time uplink type
1286     ORA     #IDLE_UPLINK  ;Set uplink idle bit and save as current type
1287     STA     curr_tlm_type
1288     LDA     TELSTAT       ;Get current value of TELSTAT
1289     AND     #TELSTAT_MSK  ; and mask changeable bits
1290     ORA     curr_tlm_type  ;Set new uplink type
1291     STA     TELSTAT       ;Write new TELSTAT and return
1292 STT_END
1293     RTS
1294 ;-----
1295 ;a END; (* SET_TLM_TYPE *)
1296 ;-----
1297
1298 SEJECT

```

01DE& C6 0000*
 01E1& A4 01
 01E3& 27 04
 01E5& 1E 00*
 01E7& 20 02
 01E9& 1F 00*
 01EB& C6 0000*
 01EE& A4 C6
 01F0& AA 01
 01F2& B7 00*
 01F4& B6 00*
 01F6& A4 38
 01F8& BA 00*
 01FA& B7 00*
 01FC& 81

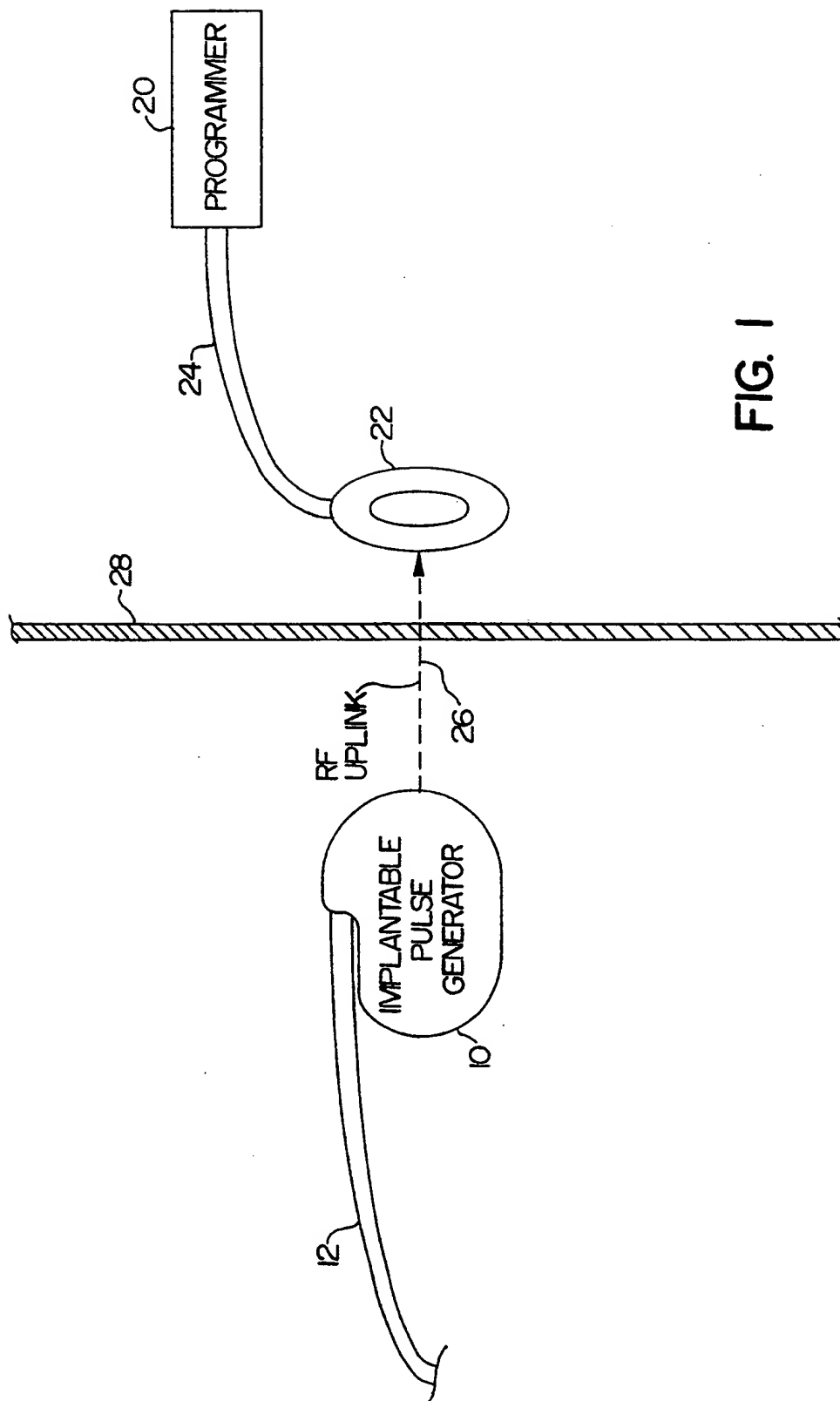
-40-

WHAT IS CLAIMED IS:

- 1 1. A method for transmitting data percutaneously
2 between a medical device implanted within a human body
3 and an external device, comprising the steps of:
4 (a) formatting the data to be transmitted by:
5 (1) establishing a frame having a fixed time
6 interval;
7 (2) placing a unique synchronizing signal at a
8 first fixed range within said frame;
9 (3) placing a frame identifier at a second
10 fixed range within said frame; and
11 (4) placing said data at a third fixed range
12 within said frame; and
13 (b) transmitting said formatted data between said
14 implanted medical device and said external
15 device.
- 1 2. A method according to claim 1, wherein said
2 data is representative of more than one type of data, and
3 wherein said frame identifier is indicative of the data
4 type within said frame being transmitted.
- 1 3. A method according to claim 2, wherein said
2 data is in digital format.
- 1 4. A method according to claim 3, wherein each of
2 said steps (a)(2), (a)(3) and (a)(4) thereof further
3 comprises generating a burst of radio frequency energy at
4 a time within the corresponding fixed range appropriate
5 to pulse position modulate said burst.
- 1 5. An apparatus for transmitting data
2 percutaneously between an implantable medical device and
3 an external device, comprising:
4 (a) frame defining means for defining a
5 transmission frame of a fixed time interval;

-41-

- 6 (b) first means coupled to said frame defining
7 means for transmitting a synchronizing signal
8 within a first time range of said transmission
9 frame;
10 (c) second means coupled to said frame defining
11 means for transmitting a frame identifier
12 within a second time range of said transmission
13 frame; and
14 (d) third means coupled to said frame defining
15 means for transmitting said data within a third
16 time range of said transmission frame.



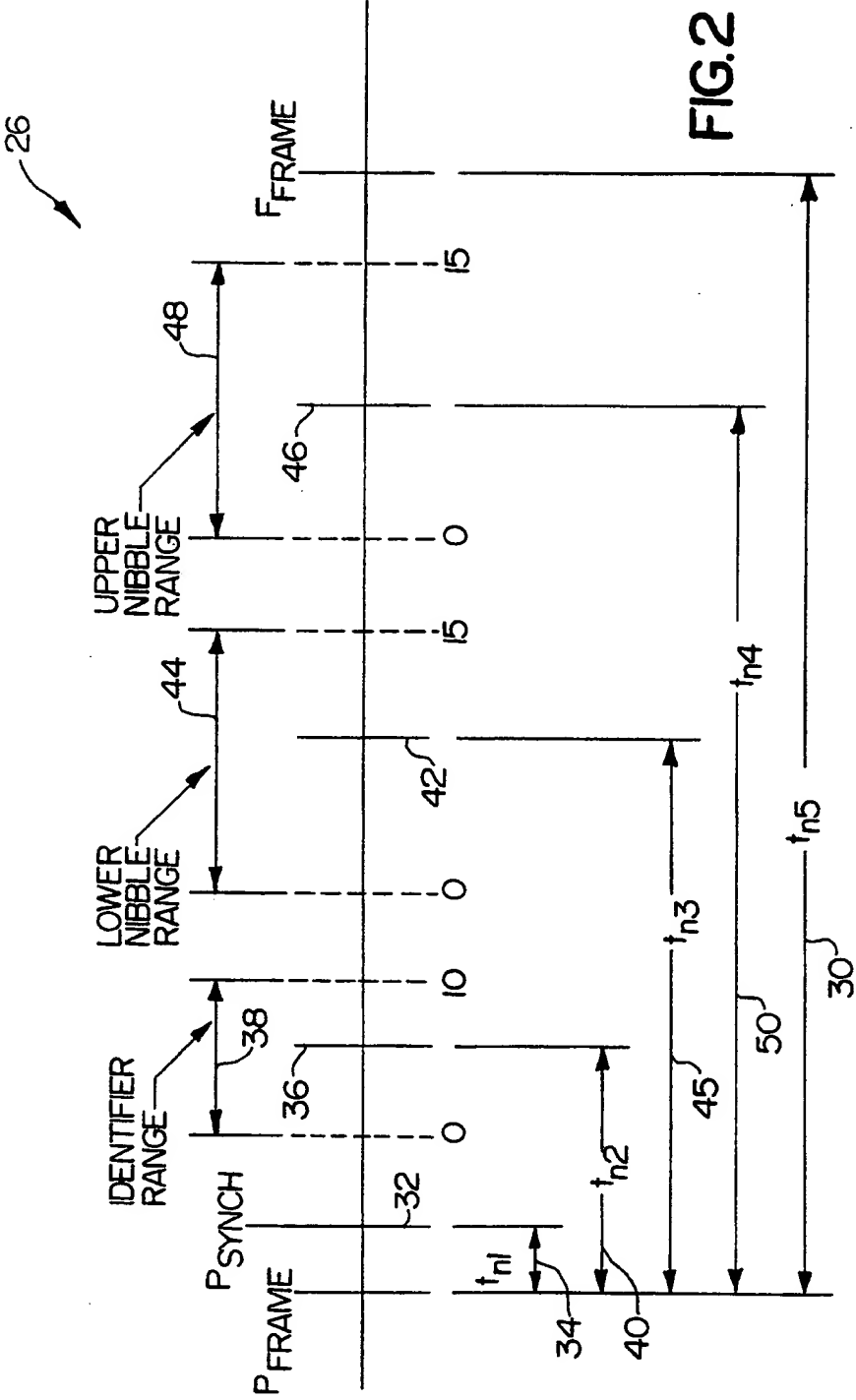
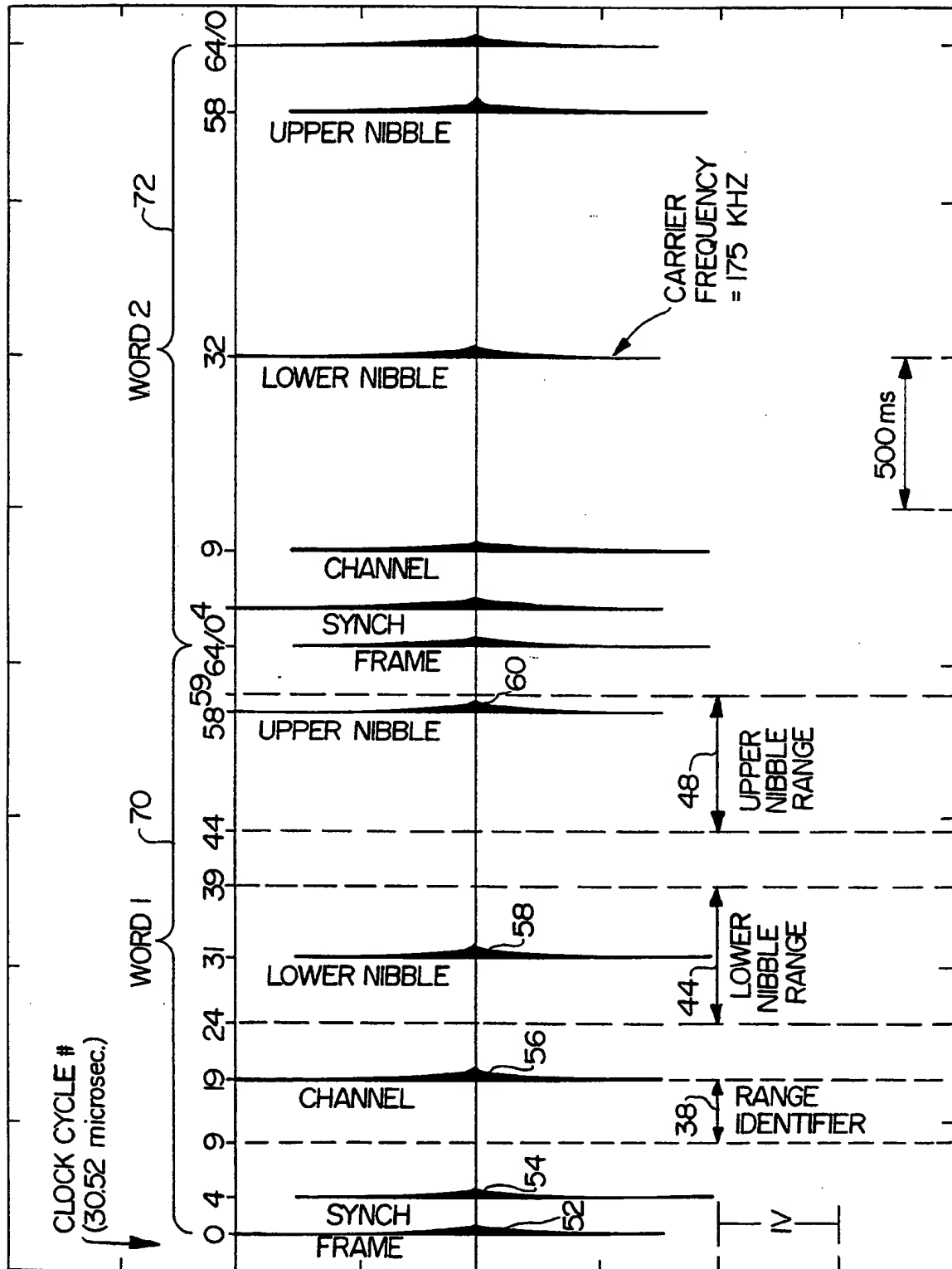


FIG. 3



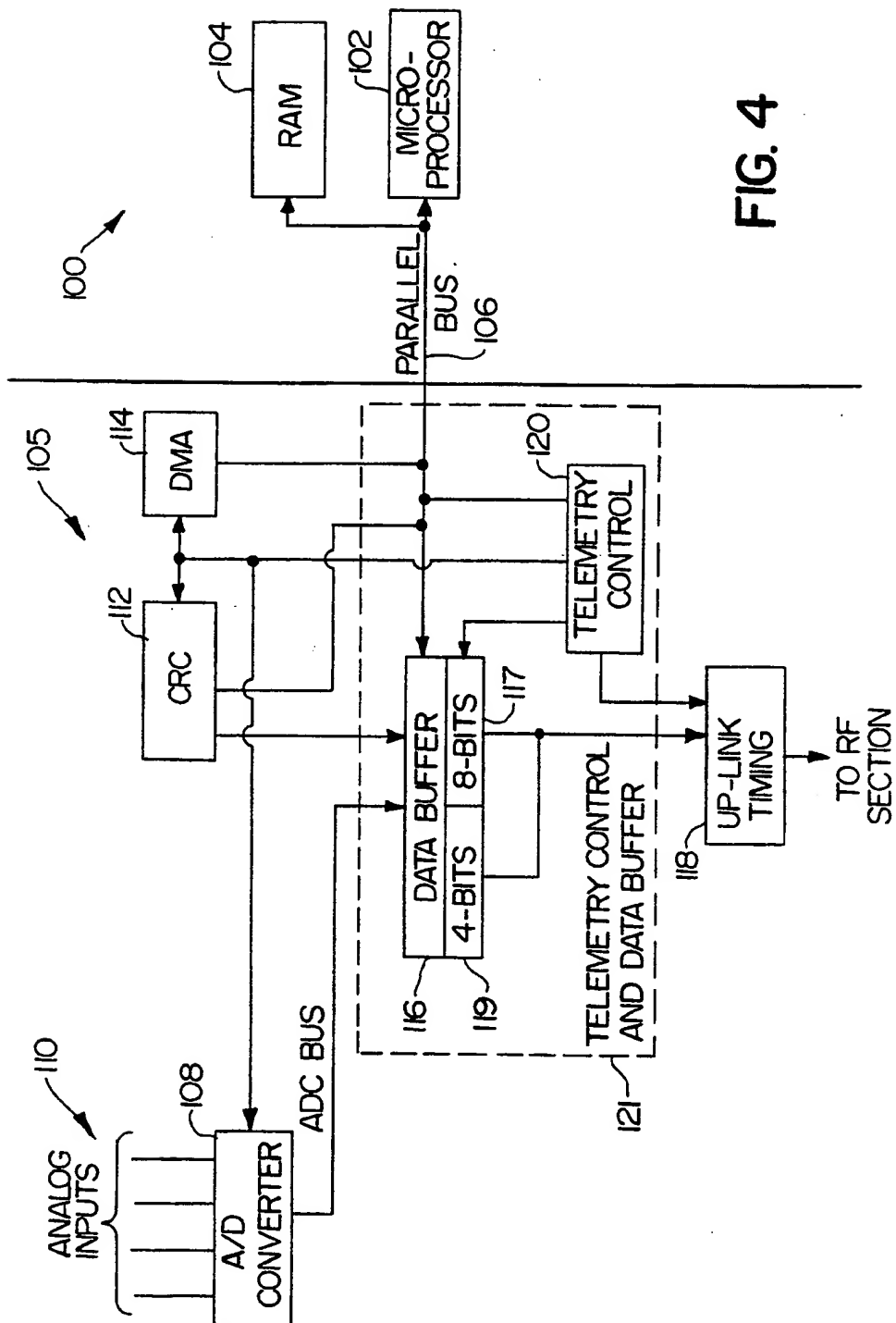


FIG. 4

5 / 8

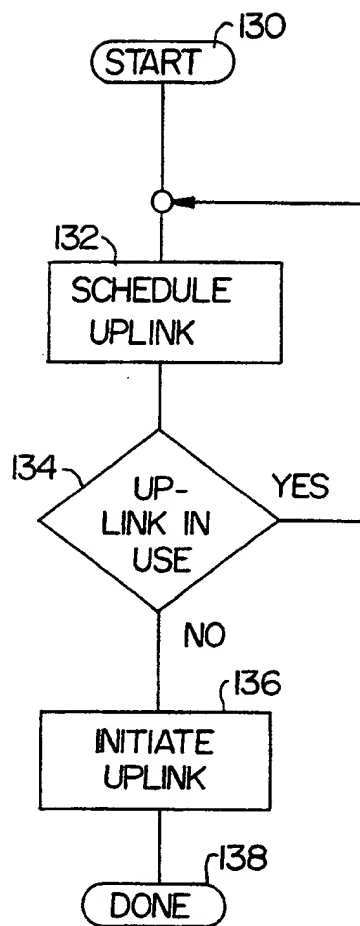


FIG. 5

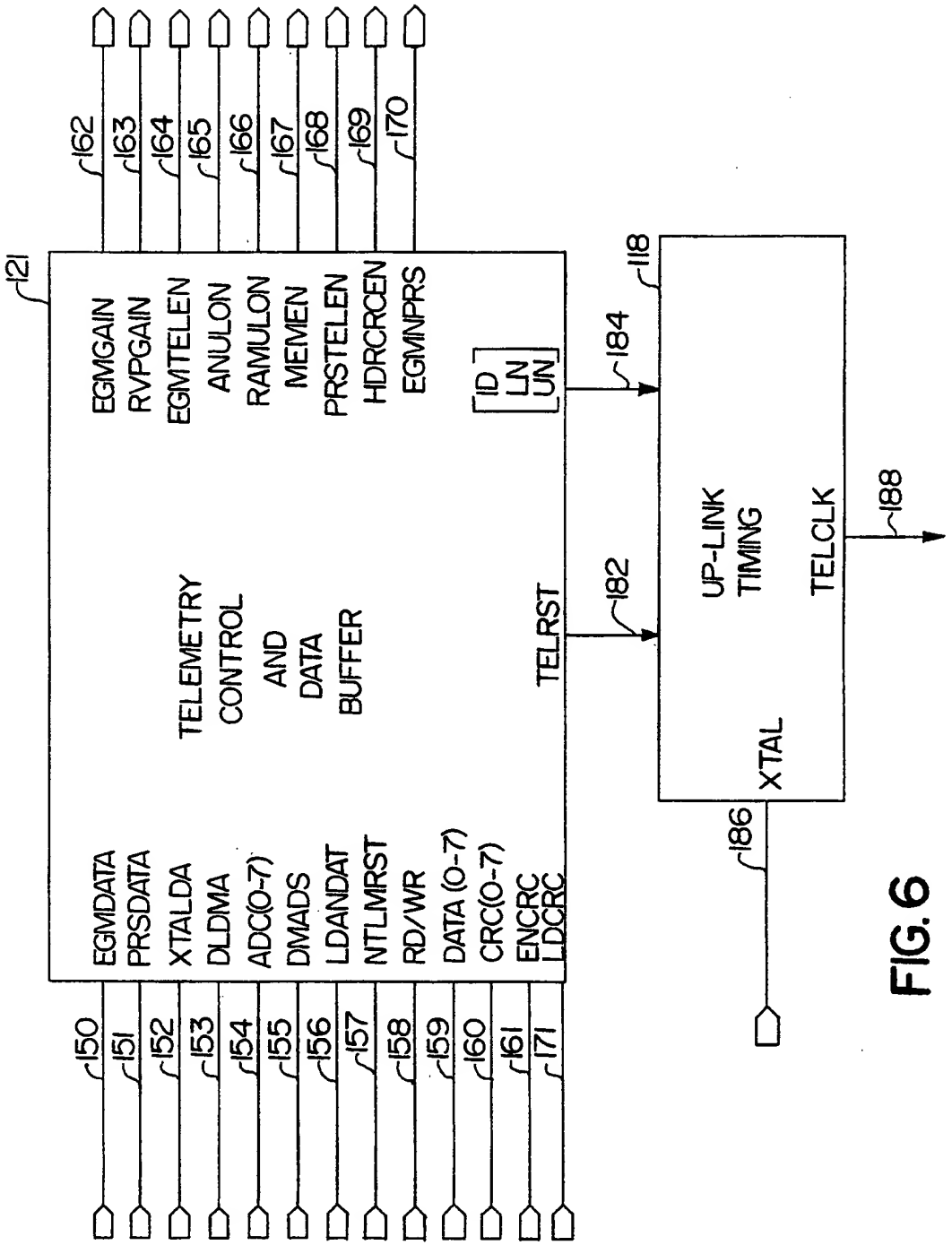
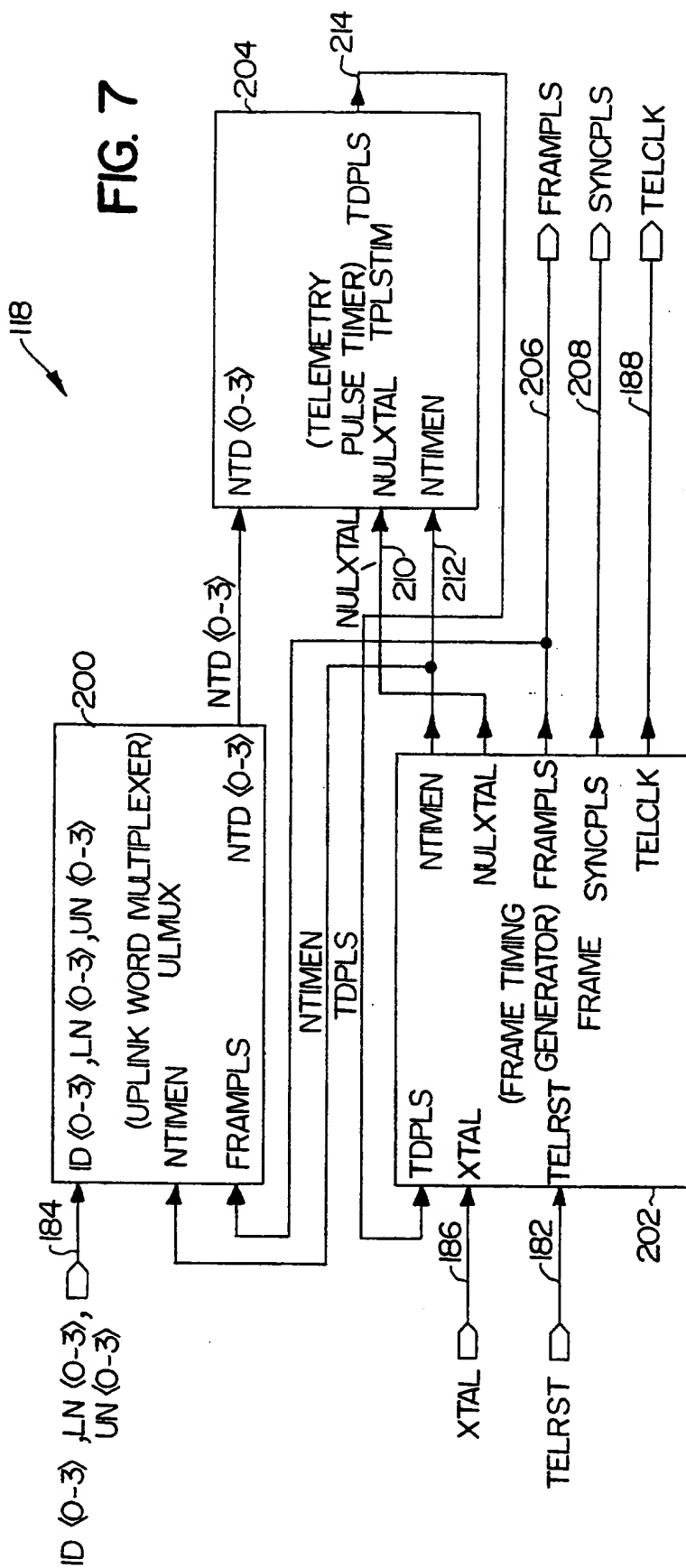


FIG. 6



**ANNEX TO THE INTERNATIONAL SEARCH REPORT
ON INTERNATIONAL PATENT APPLICATION NO. PCT/US 91/00309**

SA 44478

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report.
The members are as contained in the European Patent Office EDP file on 23/03/91
The European Patent office is in no way liable for these particulars which are merely given for the purpose of information.

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A2- 0071131	09/02/83	AU-D- 8631682	03/02/83
		DE-A- 3130104	17/02/83
		JP-A- 58041570	10/03/83
		US-A- 4524774	25/06/85
DE-C2- 2703700	04/08/83	NONE	
DE-A1- 3119119	09/12/82	NONE	
US-A- 4556063	03/12/85	CA-A- 1183576	05/03/85
		CA-C- 1187140	14/05/85
		DE-A- 3139452	24/06/82
		FR-A-B- 2491659	09/04/82
		JP-A- 57089872	04/06/82
		NL-A- 8104534	03/05/82

For more details about this annex : see Official Journal of the European patent Office, No. 12/82